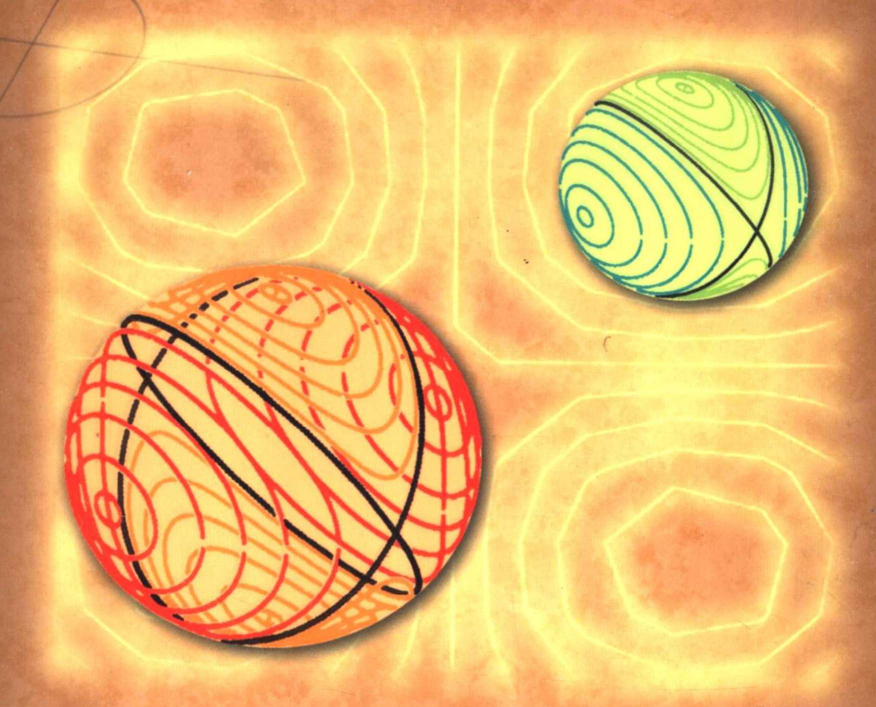
TURING 图灵数学·统计学丛书 10





Applied Numerical Linear Algebra

应用数值线性代数

[美] James W. Demmel 著

王国荣 译



人民邮电出版社
POSTS & TELECOM PRESS

应用数值线性代数

Applied Numerical Linear Algebra

"……这本书非常通俗地讲述数值线性代数，特别适合各类理工科专业一年级研究生。对舍入误差分析和扰动理论的论述透彻而详尽……作者的文笔清晰，条理分明，非常易于阅读。"

—— William W. Hager, *Mathematical Reviews*

"如果你想要从事任何涉及矩阵的计算——包括线性方程组、最小二乘法和特征值，这本书将帮助你知其然，更知其所以然。它提供了最新的资料，可作为教材或参考书……"

—— L. Ehrlich, *Computing Review*

本书全面介绍了线性方程组求解、最小二乘法、特征值、单一值的分解等问题的解决方案。作者以自己多年的教学研究经验以及开发 LAPACK 和 ScaLAPACK 的体会，提供了解决这些问题的最新的技巧和方法，并对各种具体情况下如何选择算法提供了建议，也推荐了很多应用于实际情况的算法。对于每一个问题，本书既提供了数学背景，又提供了用于计算机软件的算法，是一本非常实用的教材或参考书。

这本优秀的教材已被很多著名大学采用，包括美国麻省理工学院、斯坦福大学、加州大学伯克利分校和戴维斯分校、英国曼彻斯特大学等。



James W. Demmel 加州大学伯克利分校教授，世界著名的数值分析学家，1993年荣获SIAM（国际工程与应用数学学会）颁发的数值分析和科学计算领域最高荣誉的威尔金森奖。他还是美国工程院院士，IEEE会士，ACM会士。他领导开发的数值线性代数库LAPACK，已成为工业界的主流标准之一。

本书相关信息请访问：
图灵网站 http://www.turingbook.com
读者/作者热线：(010) 88593802
反馈/投稿/推荐信箱：contact@turingbook.com

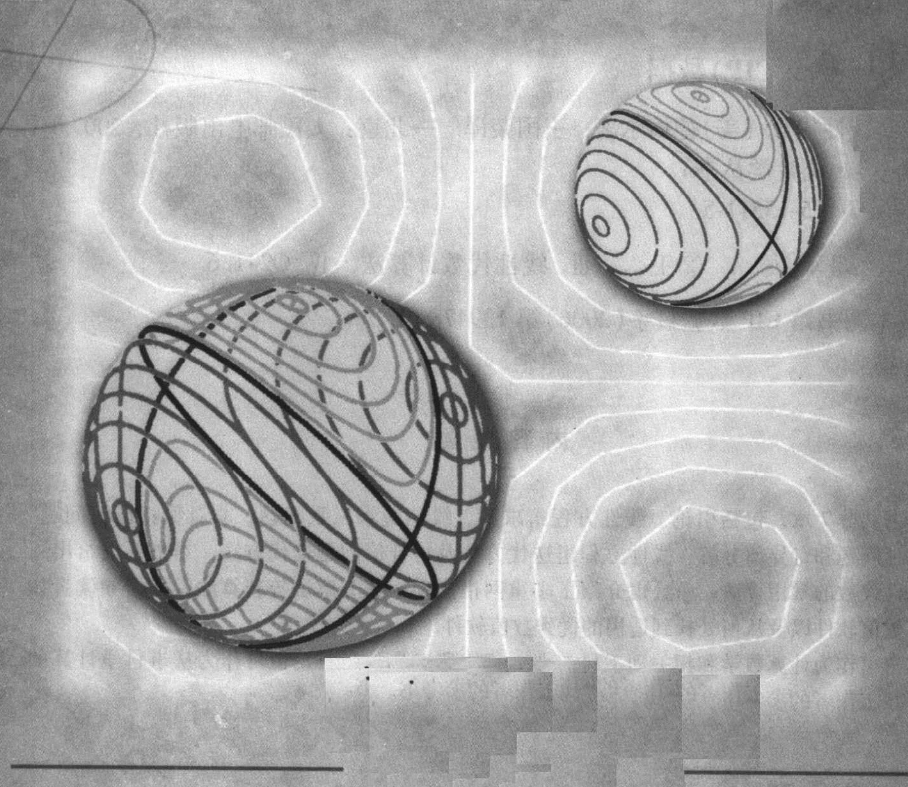
分类建议 数学/计算数学

人民邮电出版社网址 www.ptpress.com.cn



ISBN 978-7-115-15511-5/O1
定价：49.00 元

TURING 图灵数学·统计学丛书 10



Applied Numerical Linear Algebra

应用数值线性代数

[美] James W. Demmel 著

王国荣 译

人民邮电出版社

北 京

图书在版编目(CIP)数据

应用数值线性代数/(美)德梅尔著；王国荣译. —北京：人民邮电出版社，2007.6
(图灵数学·统计学丛书)
ISBN 978-7-115-15511-5

Ⅰ. 应… Ⅱ. ①德… ②王… Ⅲ. 线性代数计算法 Ⅳ. O241.6

中国版本图书馆 CIP 数据核字(2006)第 139478 号

内容提要

全书共分 7 章，包括引论、线性方程组求解、线性最小二乘问题、非对称特征值问题、对称特征问题和奇异值分解、线性方程组迭代方法及特征值问题迭代方法. 本书不仅给出了数值线性代数的常用算法，而且也介绍了多重网格法和区域分解法等新算法，并指导读者如何编写数值软件以及从何处找到适用的优秀数值软件.

本书可作为计算数学和相关理工科专业一年级研究生的教材，也可作为从事科学计算的广大科技工作者的参考书.

图灵数学·统计学丛书

应用数值线性代数

◆ 著 [美] James W Demmel
译 王国荣
责任编辑 明永玲 李 颖

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress com cn
网址 http //www ptpress com cn
北京铭成印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本：700 × 1000 1/16
印张：22 彩插 4
字数：724 千字 2007 年 6 月第 1 版
印数：1 – 4 000 册 2007 年 6 月北京第 1 次印刷

著作权合同登记号 图字：01-2006-1279 号

ISBN 978-7-115-15511-5/O1

定价：49 00 元

读者服务热线：(010)88593802 印装质量热线：(010)67129223

译 者 介 绍

王国荣 1960年上海师范学院数学系毕业后留校任教，历任上海师范大学数学科学学院院长和数学科学研究所所长，中国计算数学学会理事，中国工业与应用数学学会理事，现为上海师范大学教授，博士生导师，中国线性代数学会常务理事. 1988年和1996年两次公派赴美国Iowa大学及North Carolina州立大学作访问和合作研究. 曾3次获国家自然科学基金资助，从事广义逆理论、应用与并行算法研究. 曾获得国家教委科技进步三等奖(1994)，国务院特殊津贴(1997)，上海市优秀教育工作者(2004)，上海市科技进步三等奖(2005)，上海市级教学成果二等奖(2005).

在广义逆的扰动理论、条件数、递推算法、有限算法、嵌入算法、并行算法、Cramer法则的推广，广义逆的子式，广义逆的反序律以及算子广义逆的表示与逼近等方面取得了一系列研究成果，在国内外刊物上发表论文100篇，其中在SCI刊物LAA等上发表了25篇. 已独立或合作完成7本教材和专著的翻译及撰写工作：《矩阵计算引论》（G. W. Stewart著，王国荣等译），《数值分析引论》（K. E. Atkinson著，匡蛟勋、王国荣等译），《矩阵与算子广义逆》（王国荣著），《大学数学》（一）、（二）（王国荣主编），*Generalized Inverses: Theory and Computations*（《广义逆：理论与计算》，王国荣等），《数值分析》（D. Kincard，W. Cheney著，王国荣等译）.

译 者 序

数值线性代数是一门内容十分广泛的学科，它随着计算机和数值软件水平的不断提高发展越来越迅速. J. W. Demmel 教授所著《应用数值线性代数》（1997）是一本富有特色的、优秀的数值线性代数教材.

本书是根据作者制订的5个目标撰写的，它不仅适合计算科学和相关理工科专业一年级研究生使用，而且对广大从事科学计算的工程技术人员也有重要的参考价值；本书不仅给出线性方程组求解、线性最小二乘问题、特征值问题和奇异值分解常用的直接法和迭代法，而且介绍多重网格法、区域分解法等反映当前技术水平的新算法；此外，本书作者还强调学生不仅应该学会编写数学软件，而且能够从其他地方找出适用的优秀数值软件. 学习本书需要线性代数和程序设计方面的基础知识.

作为一本教材，本书配有大量课外作业题，涉及程序设计的问题均用“程序设计”标出，其余所有问题分为容易、中等和困难三档，便于读者深入理解和掌握有关的内容.

作者在前言中还列出本书的九大特色. 可以看出本书作者在新算法的设计和应用背景以及对各种算法的性能比较等方面都下了很大的工夫.

本人十分荣幸地应图灵公司之邀在半年时间内将本书译出，深感时间紧迫，译出后又来不及进行教学实践，因此译文中疏误和不妥之处在所难免. 敬请广大读者指正，以便再版时改正.

王国荣

2006年3月于上海师范大学

前　　言

本教材包含线性方程组求解、最小二乘问题、特征问题和奇异值分解的直接法和迭代法. 本书较早的版本自1990年起作者在加州大学伯克利分校数学系研究生班中使用过，之前曾在柯朗研究所使用过.

在本书的写作过程中，我力求实现下列目标：

1. 教材应该吸引来自各种工科和理科的一年级研究生.

2. 教材应该是自成体系的，它仅仅假定一名优秀大学生具有线性代数方面的背景知识.

3. 学生应该学到本领域的数学基础知识以及学会如何编写数值软件或者找出优秀的数值软件.

4. 学生应该学到有效地解决实际问题的实用知识. 尤其是，即使我在本书中只作了较简单的描述，他们也应该能了解每个领域中当前最新的方法，或者何时去寻找它们以及在何处去找到它们.

5. 教材应该正好适合一个学期，因为大多数学生这门课程都是一个学期.

实际上，促使我写本书的原因是目前已有的各种教材虽然非常优秀，但是不能实现上述目标. Golub 和 Van Loan 的教材[121]是百科全书式的，然而仍旧省略了某些重要的论题，例如多重网格法、区域分解法以及特征值问题的最新算法. Watkins [252]和 Trefethen 与 Bau[243]的教材也省略了某些当前最新的算法.

虽然我相信上述5个目标已实现. 但是第5个目标最难以处理. 特别为了包含最新的研究成果以及来自同事们的要求，教材渐渐变厚而超过课时. 以本书为基础的适度的一门课程应该包含：

- 第1章，1.5.1节除外；
- 第2章，2.2.1，2.4.3，2.5，2.6.3和2.6.4节除外；
- 第3章，3.5和3.6节除外；
- 第4章，直到4.4.5节，包括4.4.5节；
- 第5章，5.2.1，5.3.5，5.4和5.5节除外；
- 第6章，6.3.3，6.5.5，6.5.6，6.6.6，6.7.2，6.7.3，6.7.4，6.8，6.9.2和6.10节除外；
- 第7章，直到7.3节，包括7.3节.

本书包含下列显著特色：

- 一个课程主页，提供了教材中的例题和课外问题的 Matlab 源代码；
- 经常推荐和指出当前可利用的最佳软件(来自 LAPACK 和其他地方)；

- 讨论现代的基于高速缓冲存储器的计算机存储器是如何影响算法设计的；
- 对最小二乘问题和对称特征值问题的一些相互竞争的算法的性能进行比较；
- 讨论从雅可比法到多重网格法的各种迭代法求解正方形网格上泊松方程，并作详尽的性能比较；
- 对关于对称特征值问题的兰乔斯算法作详尽的讨论并给出数值例子；
- 从机械振动、计算几何等一些领域中提取数值例子；
- 含关于对称特征值问题和奇异值分解的“相对扰动理论”和相应的高精度算法等内容；
- 特征值算法的动力系统解释.

课程主页的 URL 为 http://www.siam.org/books/demmel/demmel_class，在本教材中简写为 HOMEPAGE. 也将使用其他两个简写的 URL. PARALLEL_HOMEPAGE 是http://www.siam.org/books/demmel/demmel_parallelclass 的简写，并指出作者在并行计算方面有关的在线的课程(on-line class). NETLIB 是 http://www.netlib.org 的简写.

课外问题按其难度用容易、中等或困难标出. 需要较多程序设计的问题用“程序设计”标出.

致谢

许多人都对本书作出了贡献. Zhaojun Bai 将本教材用于德州农机大学及肯塔基大学的教学，贡献了许多课外问题及有用的建议. Alan Edelman(用于麻省理工学院), Martin Gutknecht(用于苏黎世理工学院), Velvel Kahan(用于伯克利), Richard Lehoucq, Beresford Parlett 及许多不留姓名的人给本书提供了许多建议. 表 2-2 取自我以前的学生 Xiaoye Li 的博士论文. Mark Adams, Tzu-Yi Chen, Inderjit Dhillon, Jian Xun He, Melody Ivory, Xiaoye Li, Bernd Pfrommer, Huan Ren 和 Ken Stanley 及其他许多在柯朗、伯克利、肯塔基和麻省理工的学生在这些年来，帮我改正了文中不少的错误. Bob Untiedt 和 Selene Victor 在制图和打字方面帮了很大的忙. Megan 提供了封面照片. 最后，感谢 Kathy Yelick 多年来始终不渝地支持本书的写作.

J. W. Demmel

1997 年 6 月于加利福尼亚伯克利

目 录

第1章 引　　论

1.1 基本符号

在本书中我们将经常提及矩阵、向量和标量. 矩阵用 $\boldsymbol{A}$ 这样的大写黑体字母表示，其(i,j)元素用 a_{ij} 表示. 若矩阵用像 $\boldsymbol{A}+\boldsymbol{B}$ 这样的表达式给出，其(i,j)元素将记作 $(\boldsymbol{A}+\boldsymbol{B})_{ij}$. 在详细的算法描述中，有时将记作 $\boldsymbol{A}(i,j)$ 或利用 Matlab™ [1] [184] 符号 $\boldsymbol{A}(i:j,k:l)$ 表示 $\boldsymbol{A}$ 的位于第 i 行到第 j 行以及第 k 列到第 l 列的子阵. 像 $\boldsymbol{x}$ 这样的小写黑体字母表示一个向量，其第 i 个元素记作 x_i. 向量几乎总是列向量，它如同具有一列的矩阵. 小写的希腊字母(偶尔用小写字母)表示标量. $\mathbb{R}$ 表示实数集；$\mathbb{R}^n$ 表示 n 维实向量集；$\mathbb{R}^{m\times n}$ 表示 $m\times n$ 阶实阵集；$\mathbb{C}$、$\mathbb{C}^n$ 和 $\mathbb{C}^{m\times n}$ 分别表示复数集、复向量集和复 $m\times n$ 阶阵集. 偶尔我们用简略的表达法 $\boldsymbol{A}^{m\times n}$ 表示 $\boldsymbol{A}$ 是 $m\times n$ 阶阵. $\boldsymbol{A}^{\mathrm{T}}$ 表示矩阵 $\boldsymbol{A}$ 的转置：$(\boldsymbol{A}^{\mathrm{T}})_{ij}=a_{ji}$. 对复矩阵也使用共轭转置 $\boldsymbol{A}^*$：$(\boldsymbol{A}^*)_{ij}=\bar{a}_{ji}$. $\Re z$ 和 $\Im z$ 分别表示复数 z 的实部和虚部. 若 $\boldsymbol{A}$ 是 $m\times n$ 阶阵，则 $|\boldsymbol{A}|$ 是 $\boldsymbol{A}$ 的元素的绝对值构成的 $m\times n$ 阶阵：$(|\boldsymbol{A}|)_{ij}=|a_{ij}|$. $|\boldsymbol{A}|\leqslant|\boldsymbol{B}|$ 的不等式表示：对一切 i 和 j，$|a_{ij}|\leqslant|b_{ij}|$. 我们也对向量用这个绝对值符号：$(|\boldsymbol{x}|)_i=|x_i|$. 证明的结束将用□来标记，例子的结束用◇来标记. 其他符号将在需要时介绍.

1.2 数值线性代数的标准问题

1

我们将考虑下列标准问题：

- 线性方程组：解 $\boldsymbol{Ax}=\boldsymbol{b}$. 这里 $\boldsymbol{A}$ 是一个已知的 $n\times n$ 阶非奇异实的或复的矩阵，$\boldsymbol{b}$ 是一个已知的 n 维列向量，$\boldsymbol{x}$ 是我们要求解的 n 维列向量.
- 最小二乘问题：计算极小化 $\|\boldsymbol{Ax}-\boldsymbol{b}\|_2$ 的 $\boldsymbol{x}$，这里 $\boldsymbol{A}$ 是 $m\times n$ 阶的，$\boldsymbol{b}$ 是 $m\times 1$ 阶的，$\boldsymbol{x}$ 是 $n\times 1$ 阶的，而 $\|\boldsymbol{y}\|_2\equiv\sqrt{\sum_i|y_i|^2}$ 称为向量 $\boldsymbol{y}$ 的 2-范数. 若 $m>n$，即方程数大于未知量的个数，这个方程组称为*超定的*. 此时，一般不能精确地求解 $\boldsymbol{Ax}=\boldsymbol{b}$. 若 $m<n$，这个方程组称为*亚定的*，其将有无穷多个解.
- 特征值问题：给定 $n\times n$ 阶矩阵 $\boldsymbol{A}$，求 $n\times 1$ 阶非零向量 $\boldsymbol{x}$ 和标量 λ 使得 $\boldsymbol{Ax}=\lambda\boldsymbol{x}$.

---

1. Matlab 是 MathWorks 股份有限公司的注册商标.

- 奇异值问题：给定 $m \times n$ 阶矩阵 $\boldsymbol{A}$，求 $n \times 1$ 阶非量向量 $\boldsymbol{x}$ 和标量 λ 使得$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$，我们将看到这个特殊类型的特征值问题是足够重要的，值得专门考虑这个问题及其算法.

之所以将上述这些标准问题选出来加以强调，是因为它们如此频繁地在工程和科学实践中出现. 本书将通过从工程、统计和其他领域中提炼出来的简单例子来说明它们. 还有很多这些标准问题的变形我们也将予以考虑，诸如广义特征值问题 $\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{B}\boldsymbol{x}$(4.5 节)和“秩亏”最小二乘问题 $\min_x \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$，因为 $\boldsymbol{A}$ 的列线性相关，所以其解不唯一(3.5 节).

我们将认识到利用问题中可能有的任何特殊结构是极其重要的. 例如，如果利用最一般形式的高斯消元法求解 $n \times n$ 线性方程组所付出的代价是 $2/3n^3$ 次浮点运算. 若方程组是对称正定的，则可以用另一个称为楚列斯基的算法而节省一半工作量. 若进一步知道矩阵是带状的有半带宽$\sqrt{n}$(即当 $|i-j| > \sqrt{n}$时，$a_{ij} = 0$)，则可利用带状楚列斯基(Cholesky)算法进一步减少运算次数至 $O(n^2)$阶. 若我们试图用 5 点差分近似求解正方形区域上的泊松方程，因其方法几乎唯一地确定矩阵，则利用多重网格算法可使运算次数减至 $O(n)$阶，在每个解分量正好用一个固定的工作量的意义下，这个速度几乎是最快的(6.4 节).

1.3 一般的方法

以下是几个我们将要反复使用的一般的概念和方法：

1. 矩阵分解；
2. 扰动理论和条件数；
3. 舍入误差对算法的影响，包括浮点运算的性质；
4. 分析算法的速度；
5. 数值计算软件.

下面将对这些方法逐一作简短的讨论.

1.3.1 矩阵分解

矩阵 $\boldsymbol{A}$ 的分解是把 $\boldsymbol{A}$ 表成几个“较简单”的矩阵之积，它使所讨论的问题容易求解. 我们给出两个例子.

例 1.1 假如要求解 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. 若 $\boldsymbol{A}$ 是下三角阵，利用向前回代：

for $i = 1$ to n

$$x_i = (b_i - \sum_{k=1}^{i-1} a_{ik} x_k) / a_{ii}$$

end for

容易求解

$$\begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

若 $\boldsymbol{A}$ 是上三角阵，可利用类似的向后回代思想求解. 为利用这些思想求解一般的方程组 $\boldsymbol{Ax}=\boldsymbol{b}$，需要下面的矩阵分解，它正好是高斯消元法的另一种叙述方式. ◇

定理 1.1 若 $n\times n$ 阶矩阵 $\boldsymbol{A}$ 非奇异，则存在一个置换阵 $\boldsymbol{P}$(对单位阵作行置换后得到的矩阵)、一个非奇异下三角阵 $\boldsymbol{L}$ 和一个非奇异上三角阵 $\boldsymbol{U}$，使得 $\boldsymbol{A}=\boldsymbol{P}\cdot\boldsymbol{L}\cdot\boldsymbol{U}$. 为求解 $\boldsymbol{Ax}=\boldsymbol{b}$，我们如下求解等价的方程组 $\boldsymbol{PLUx}=\boldsymbol{b}$：

$$\boldsymbol{LUx}=\boldsymbol{P}^{-1}\boldsymbol{b}=\boldsymbol{P}^{\mathrm{T}}\boldsymbol{b} \quad (\text{置换 } \boldsymbol{b} \text{ 的元素}),$$

$$\boldsymbol{Ux}=\boldsymbol{L}^{-1}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{b}) \quad (\text{向前回代}),$$

$$\boldsymbol{x}=\boldsymbol{U}^{-1}(\boldsymbol{L}^{-1}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{b}) \quad (\text{向后回代}).$$

我们将在 2.3 节中证明此定理.

例 1.2 若尔当(Jordan)典范分解 $\boldsymbol{A}=\boldsymbol{VJV}^{-1}$ 显示 $\boldsymbol{A}$ 的特征值和特征向量. 这里 $\boldsymbol{V}$ 是一个非奇异阵，其列包含特征向量，而 $\boldsymbol{J}$ 是 $\boldsymbol{A}$ 的若尔当典范型，这是一个特殊的三角阵，$\boldsymbol{A}$ 的特征值在它的对角线上. 应该认识到舒尔(Schur)分解 $\boldsymbol{A}=\boldsymbol{UTU}^*$ 数值计算上是出众的，其中 $\boldsymbol{U}$ 是酉阵(即 $\boldsymbol{U}$ 的列是规范正交的)，$\boldsymbol{T}$ 是上三角阵，$\boldsymbol{A}$ 的特征值在其对角线上. 舒尔型 $\boldsymbol{T}$ 可以比若尔当型 $\boldsymbol{J}$ 更快和更精确地算出. 我们在 4.2 节中讨论若尔当分解和舒尔分解. ◇

3

1.3.2 扰动理论和条件数

由数值算法产生的结果很少是完全正确的. 存在两个误差源. 首先，输入数据到算法中可能产生的误差，它由先前的计算或者多半是测量误差引起的. 其次，由于在算法之内作近似，所以存在由算法本身引起的误差. 为了估计从这两个误差源在计算的结果中的误差，需要推断当输入数据出现小扰动时，问题的解有多少改变(或扰动).

例 1.3 设 $f(x)$ 是实变量 x 的一个实值可微函数. 我们要计算 $f(x)$，但不知道确切的 x. 假定代之给定 $x+\delta x$ 和 δx 的一个界，我们所能做的(在没有更多的信息时)就是计算 $f(x+\delta x)$，并试图给出绝对误差 $|f(x+\delta x)-f(x)|$ 的界. 可以利用一个简单的对 f 的线性近似得到估计 $f(x+\delta x)\approx f(x)+\delta x f'(x)$，故误差是 $|f(x+\delta x)-f(x)|\approx|\delta x|\cdot|f'(x)|$. 称 $|f'(x)|$ 为 f 在 x 上的绝对条件数. 若 $|f'(x)|$ 足够大，则即使 δx 是小的，误差可能是大的，此时，我们称 f 于 x 处

病态. ◇

我们称之为绝对条件数是因为给定输入值的绝对变化 $|\delta x|$ 的一个界时，它提供绝对误差 $|f(x+\delta x)-f(x)|$ 的一个界. 通常也使用下列本质上等价的表达式来界定误差：

$$\frac{|f(x+\delta x)-f(x)|}{|f(x)|}\approx\frac{|\delta x|}{|x|}\cdot\frac{|f'(x)|\cdot|x|}{|f(x)|}.$$

这个表达式界定相对误差 $|f(x+\delta x)-f(x)|/|f(x)|$ 针对输入时相对改变 $|\delta x|/|x|$ 的倍数关系，因子 $|f'(x)|\cdot|x|/|f(x)|$ 称为相对条件数，或者简称为条件数.

当需要推断输入数据中的误差是如何影响计算结果时，条件数是头等重要的. 我们用输入误差的界乘以条件数简单地去界定计算解中的误差.

4 对每个考虑的问题，我们将导出其相应的条件数.

1.3.3 舍入误差对算法的影响

为继续分析由算法本身引起的误差，需要研究算术运算中舍入误差的影响，或简称为舍入影响. 我们将对最优良的算法所拥有的性质(向后稳定性)进行讨论. 它的定义如下：

设 $\mathrm{alg}(x)$ 是含有舍入影响的 $f(x)$ 的算法. 若对一切 x 存在一个"小的" δx 使得 $\mathrm{alg}(x)=f(x+\delta x)$，则称 $\mathrm{alg}(x)$ 为 $f(x)$ 的*向后稳定算法*，δx 称为*向后误差*. 简略地说，我们对一个有一点错误的问题 $(x+\delta x)$ 得到一个精确的解 $(f(x+\delta x))$.

这蕴含可界定误差为绝对条件数 $|f'(x)|$ 与向后误差 $|\delta x|$ 的值的乘积：

$$\text{error}=|\mathrm{alg}(x)-f(x)|=|f(x+\delta x)-f(x)|\approx|f'(x)|\cdot|\delta x|.$$

因此，若 $\mathrm{alg}(\cdot)$ 是向后稳定的，因 $|\delta x|$ 总是小的，所以，除非绝对条件数 $|f'(x)|$ 大，误差将是小的. 因而，向后稳定性是算法的一个理想的性质，而我们提出的大多数算法将总是向后稳定的. 结合相应的条件数，我们所有的计算解将有误差界.

证明一个算法是向后稳定的，需要了解机器的基本浮点运算的舍入误差，以及这些误差如何经过算法传播. 这在 1.5 节中讨论.

1.3.4 分析算法的速度

在选择求解一个问题的算法中，人们当然必须考虑它的运算速度(也称为性能)以及向后稳定性. 有好几种方法估计速度. 给定一个特定的问题实例、一个算法的特定的执行以及一台特定的计算机，人们当然可以简单地运行算法并看看它需花费多长时间. 这样做可能是困难的或耗费时间的，故通常需要作一个比较简单的估计. 实际上，一个特定的算法执行之前，原则上我们要估计它要多少时间.

估计一个算法所花时间通常的方式是计算算法所执行的 flops 或称浮点运算量. 我们将对给出的所有算法做此项工作. 然而, 在现代计算机体系结构上这通常是一个使人误解的时间估计, 因为它把计算机内部的数据转移到做乘法的地方比它实际执行乘法所用的时间可能多得多, 这在并行计算机上尤为正确, 而在常规的机器如工作站和个人计算机(PC)上也是正确的. 例如, 在 IBM RS6000/590 工作站上通过仔细地重排标准算法的运算(并利用正确的编译优化), 矩阵乘法能从 65 Mflops(每秒 100 万次浮点运算)加速到 240 Mflops, 几乎快 4 倍. 我们将在 2.6 节中进一步讨论.

5

若算法是迭代的, 即产生一系列收敛到解的近似, 而不是在某一个固定的步数后停止, 则我们必须问, 为把误差降到可忍受的水平需要多少步数. 为此, 当它收敛时需要确定是否是线性的 (即在每一步用一个常数因子 $0<c<1$ 来控制误差, 使得 $|\mathrm{error}_i|\leqslant c|\mathrm{error}_{i-1}|$ 或更快些, 例如二次的 ($|\mathrm{error}_i|\leqslant c|\mathrm{error}_{i-1}|^2$). 若两个算法都是线性的, 则可以问哪个算法有较小的常数 c. 迭代线性方程解算器及它们的收敛性分析是本书第 6 章的主题.

1.3.5 数值计算软件

在设计或选择一个数值计算软件时有三个主要问题需要考虑: 易操作性、可靠性和速度. 在本书中讨论的大多数算法设计时已经仔细地考虑了这三个问题. 如果已有软件能够解决你的问题, 则它容易操作的好处可能胜过任何其他考虑, 诸如速度. 实际上, 如果你只是偶尔利用软件来解决你的问题, 则使用由专家编写的用于一般用途的软件是比较方便的, 用不着自己编写更特殊的程序.

利用其他专家的软件有三种方式. 第一种方式是传统的软件库, 它由求解一组固定问题(如解线性方程组、求特征值等)的子程序汇集组成. 具体而言, 我们将讨论 LAPACK 库[10], 这是一组体现当前技术水平的、以 Fortran 和 C 语言编写的程序. 这个库以及其他许多像它一样的库不受版权限制, 是可以免费使用的, 见万维网上的 NETLIB[1]. LAPACK 可靠且速度快(例如, 如上所述很好地使用矩阵乘法), 但是需要注意数据结构和用户的调用次序. 在本书中对这样的软件将不断提供指示性信息.

第二种方式能提供比像 LAPACK 这样的库更加轻松的使用环境, 但也失去了某些性能. 例如商用系统 Matlab[184]或其他系统. Matlab 提供一个简单的交互式的程序设计环境, 其中所有的变量用矩阵表示(标量是 1×1 矩阵), 大部分的线性代数运算都设计为固定的函数是可利用的. 例如, "$\boldsymbol{C}=\boldsymbol{A}*\boldsymbol{B}$" 把矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的积存放在 $\boldsymbol{C}$ 中, 而 "$\boldsymbol{A}=\mathrm{inv}(\boldsymbol{B})$" 把矩阵 $\boldsymbol{B}$ 的逆存放在 $\boldsymbol{A}$ 中. 在 Matlab 中快速建立原型算法以

1. 在教材中把 URL 前缀 http://www.netlib.org 简写为 NETLIB.

[6] 及监控它们如何工作是容易的. 但是对用户来说, 因为 Matlab 自动地作了较多算法上的决定, 所以它可能不如用库程序快.

第三种方式是用简单的程序块组成较复杂算法方法, 也称模板. 当有大量的方法构造算法但对一个特殊的输入问题没有简单的法则选择最佳构造时, 可以使用模板. 所以, 许多构造必须留给用户. 这方面的一个例子可以在线性方程组解的模板: 迭代法的模块[24]中找到. 类似的一组关于特征问题的模板目前正在构造当中.

1.4 例: 多项式求值

我们用多项式求值的例子

$$p(x) = \sum_{i=0}^{d} a_i x^i$$

来说明扰动理论、条件数、向后稳定性和舍入误差分析的思想.

如下是多项式求值的霍纳(Horner)法则:

$p = a_d$

for $i = d - 1$ down to 0

  $p = x * p + a_i$

end for

对 $p(x) = (x-2)^9 = x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512$ 应用这个法则. 在图 1-1 的底部, 我们看到接近于零点 $x = 2$ 时, 由霍纳法则计算的 $p(x)$ 的值是相当难以预测的, 并有理由称之为"噪声". 图 1-1 上部展示了一条准确的曲线图.

为理解此图的含义, 观察一下当我们试图用基于对分法求单根的定位程序去求解 $p(x)$ 的零点时会发生什么情况. 对分法将在下面算法 1.1 中指出.

下面简要介绍一下对分法. 首先寻找一个区间 $[x_{low}, x_{high}]$, 在这个区间上 $p(x)$ 同时能取到正值和负值 $[p(x_{low}) \cdot p(x_{high}) < 0]$, 所以 $p(x)$ 在该区间上必有一个零点. 然后对区间中点 $x_{mid} = (x_{low} + x_{high})/2$ 计算 $p(x_{mid})$, 并观察在下半区间 $[x_{low}, x_{mid}]$ 中或者在上半区间 $[x_{mid}, x_{high}]$ 中 $p(x)$ 是否变号. 任何一种情形, 我们都可找到一个包含 $p(x)$ 一个零点且长度为原区间一半的新区间. 如此继续对分直到区间如要求的那样短.

决定选择上半区间或下半区间与 $p(x_{mid})$ 的符号有关. 考察图 1-1 底部 $p(x)$ 的

[7] 图像, 可以看到当 x 变化时, 其符号迅速地从正变到负, 故 x_{low} 和 x_{high} 稍稍的改变就可能完全改变决定符号的结论以及最后的区间. 实际上这与 x_{low} 和 x_{high} 的初始选择有关, 从 1.95 到 2.05 "噪声区域"内部无论何处出发算法都能收敛(见问题 1.21).

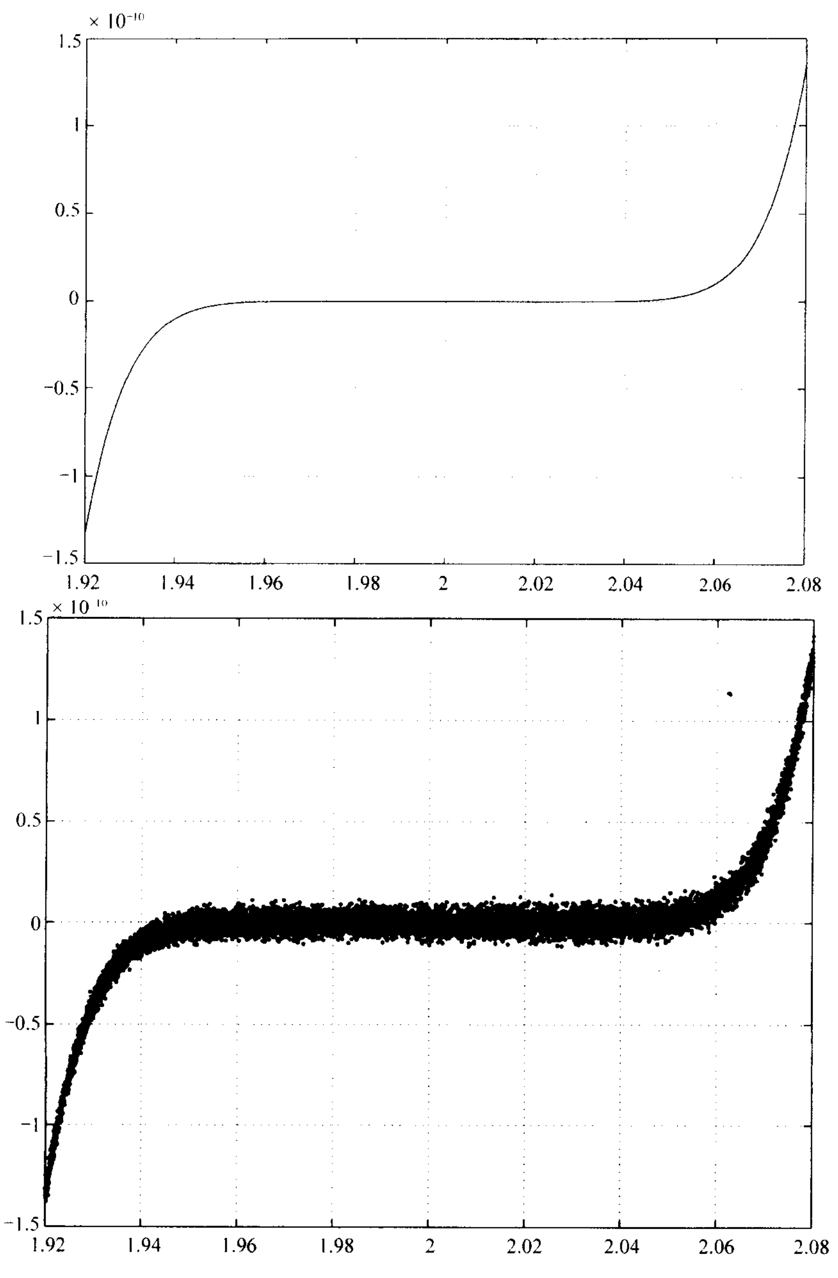


图 1-1 分别利用 $y=(x-2)^9$（上部）和霍纳法则（底部）在 8000 个等距点上求值作 $y=(x-2)^9=x^9-18x^8+144x^7-672x^6+2016x^5-4032x^4+5376x^3-4608x^2+2304x-512$ 的图形

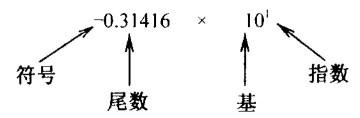
算法 1.1 利用对分法求 $p(x)$ 的零点.

```
proc bisect (p, x_low, x_high, tol)
/* find a root of p(x) = 0 in [x_low, x_high]
     assuming p(x_low) · p(x_high) < 0 */
/* stop if zero found to within ± tol */
p_low = p(x_low)
p_high = p(x_high)
while x_high - x_low > 2 · tol
     x_mid = (x_low + x_high)/2
     p_mid = p(x_mid)
     if p_low · p_mid < 0 then /* there is a root in[x_low, x_mid] */
          x_high = x_mid
          p_high = p_mid
     else if p_mid · p_high < 0 then /* there is a root in[x_mid, x_high] */
          x_low = x_mid
          p_low = p_mid
     else /* x_mid is a root */
          x_low = x_mid
          x_high = x_mid
     end if
end while
root = (x_low + x_high)/2
```

为了充分地说明这些，我们转到浮点算术运算的性质.

1.5 浮点算术运算

数 -3.1416 按科学记数法可表成如下形式：



计算机使用一个类似的称为浮点的表达式，但是一般说来基是 2.（IBM 370 的基为 16，某些电子制表软件和大多数计算器的基为 10.）例如，$0.10101_2 \times 2^3 = 5.25_{10}$.

若尾数的首位数字非零时，浮点数称为规格化的. 例如，$0.10101_2 \times 2^3$ 是规格化的，而 $0.010101_2 \times 2^4$ 不是规格化的. 浮点数通常是规格化的，它有两个好处. 每个非零浮点数作为一个位字符串，有一个唯一的表达式；并且在二进制中尾数中的首位 1 不需明确地存储(因为它总是 1)，从而为一个较长的更准确的尾数留一个额外的位.

9

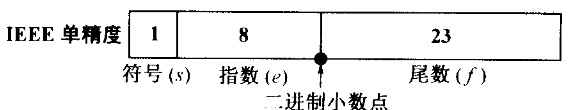
描述浮点数最重要的参数是基. 在尾数中数字个数(位)确定精度；在指数中的数字个数(位)确定指数的范围，从而确定可表示的最大的数和最小的数. 不同的浮点算术运算在下列几个方面提供的结果是不同的：它们如何舍入计算结果，它们如何处理太接近于零的数(下溢)或太大的数(上溢)，是否允许 $\pm\infty$ 以及是否使用非数(有时也称为 NaN，不定数，或保留操作数). 下面我们讨论这些情况.

首先，考虑用什么数来表示精度. 例如，0.31416×10^1 有 5 个 10 进位数字，故所有小于 0.5×10^{-4} 的信息可能被丢失. 这意味着若 x 是一个实数，其最佳的 5 位数字近似是 0.31416×10^1，则在 0.31416×10^1 中的相对表示误差是

$$\frac{|x - 0.31416 \times 10^1|}{0.31416 \times 10^1} \leqslant \frac{0.5 \times 10^{-4}}{0.31416 \times 10^1} \approx 0.16 \times 10^{-4}.$$

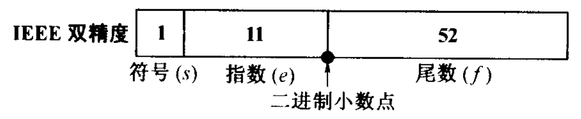
在规格化数中的最大相对表示误差是 0.10000×10^1，这是从 0.999995 到 1.00005 区间中所有数的最准确 5 位数近似. 所以它的相对误差以 0.5×10^{-4} 为界. 更一般地，在一个 p 位数字和基为 β 的浮点运算中的最大相对表示误差是 $0.5 \times \beta^{1-p}$，这也是 1 和下一个最大的浮点数 $1 + \beta^{1-p}$ 之间距离的一半.

计算机自发明以来，曾选用过许多不同的基、数字位数和取值范围，但现在最通用的是二进制算术运算的 IEEE 标准. 它被用于 Sun，DEC，HP 和 IBM 工作站以及所有的个人计算机. IEEE 算术运算包括两类浮点数：单精度(32 位字长)和双精度(64 位字长).



在 IEEE 单精度格式中，若 s，e 和 $f<1$ 分别是 1 位符号，8 位指数和 23 位尾数，则表示的数是 $(-1)^s \cdot 2^{e-127} \cdot (1+f)$. 最大的相对表示误差是 $2^{-24} \approx 6 \cdot 10^{-8}$，正的规格化数取值范围是从 2^{-126}(下溢阈值)到 $2^{127} \cdot (2-2^{-23}) \approx 2^{128}$(上溢阈值)，或者大约从 10^{-38} 到 10^{38}. 在实数系上这些浮点数的位置如图 1-2 所示. (为表示简单起见，这里我们利用 3 位尾数.)

10



在 IEEE 双精度格式中若 s，e 和 $f<1$ 分别是 1 位符号、11 位指数和 52 位尾数，则表示的数是 $(-1)^s \cdot 2^{e-1023} \cdot (1+f)$. 最大的相对表示误差是 $2^{-53} \approx 10^{-16}$，指数取

值范围是从 2^{-1022}(下溢阈值)到 $2^{1023}\cdot(2-2^{-52})\approx 2^{1024}$(上溢阈值)，或大约从 10^{-308} 到 10^{308}.

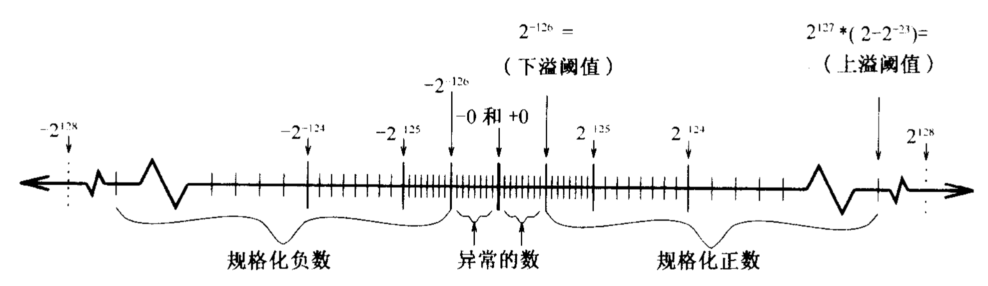


图1-2 有浮点数的实数采用实刻度线表示. 对 IEEE 单精度该取值范围是正确的，但是为表示简单起见，假定3位尾数使得在连续的2的幂之间只存在 $2^3-1=7$ 个浮点数而不是 $2^{23}-1$ 个浮点数. 连续的刻度线间的距离是2的幂和2的幂的双倍/对半之间的常数(规格化浮点数之间). $+2^{128}$ 和 -2^{128} 是在最后的位置上的一个单位，量值上比上溢阈值更大(最大的有限浮点数是 $2^{127}\cdot(2-2^{-23})$). 它们用虚刻度线表示. 图形关于0对称. +0 和 -0 是不同的浮点位字符串，但数值上相等. 用零作除法是对不同符号的零参数给出不同的结果 $+\infty$ 和 $-\infty$ 仅有的二进制运算

当计算 $a\odot b$(这里 $\odot$ 是四种二进制算术运算 +， -， * 和/之一)的正确值时不能精确地表示成一个浮点数，在它被存放到存储器或寄存器之前，必须用一个邻近的浮点数来近似. 用 $\mathrm{fl}(a\odot b)$ 表示这个近似. 差 $(a\odot b)-\mathrm{fl}(a\odot b)$ 被称为*舍入误差*. 若 $\mathrm{fl}(a\odot b)$ 是一个最接近于 $a\odot b$ 的浮点数，则称运算*正确地舍入*(或就称舍入). IEEE 算术运算具有这个引人注目的性质. [当 $a\odot b$ 正好位于两个相伴的浮点数中间一半时，IEEE 算术运算打破僵局，选择 $\mathrm{fl}(a\odot b)$ 使其最低的有效位为零. 这称为*舍入到最接近的偶数*.] 当正确地舍入时，若 $a\odot b$ 是在指数的取值范围内(否则上溢或下溢)，则我们记

11

$$\mathrm{fl}(a\odot b)=(a\odot b)(1+\delta), \tag{1.1}$$

其中 $|\delta|$ 以 ε 为界. 称 ε 为机器 ε、机器精度或 macheps. 因为我们尽可能正确地舍入，所以 ε 等于最大相对表示误差 $0.5\cdot\beta^{1-p}$. IEEE 运算也保证 $\mathrm{fl}(\sqrt{a})=\sqrt{a}(1+\delta)$，$|\delta|\leq\varepsilon$. 这是舍入误差分析最通用的模型，将在本书中使用. 一个几乎同样的公式应用于复浮点运算，见问题1.12. 然而，公式(1.1)忽略了某些有趣的细节.

更多的细节

IEEE 算术运算也包括*异常的数*，即具有最小可能指数的非规格化浮点数. 这些代表零和最小规格化浮点数之间的很小的数，可参见图1-2. 因为下溢，这些数的出现意味着差 $\mathrm{fl}(x-y)$ 决不会是零，从而得到引人注目的性质：$x=y$ 成立当且仅当 $\mathrm{fl}(x-y)=0$. 把下溢引起的误差合并到公式(1.1)中得到

$$\mathrm{fl}(a\odot b)=(a\odot b)(1+\delta)+\eta,$$

其中 $|\delta|\leq\varepsilon$ 如前，而 $|\eta|$ 以下溢引起最大误差的一个很小的数为界(在 IEEE 单精度中是 $2^{-150}\approx10^{-45}$，在 IEEE 双精度中是 $2^{-1075}\approx10^{-324}$).

IEEE 算术运算包括符号 $\pm\infty$ 和 NaN(非数). 当运算上溢时获得 $\pm\infty$，并且按下列运算规则运转：对任意有限的浮点数 x，$x/\pm\infty=0$，对任何非零浮点数 x，$x/0=\pm\infty$，$+\infty+\infty=+\infty$ 等. 诸如 $\infty-\infty$，$\frac{\infty}{\infty}$，$\frac{0}{0}$，$\sqrt{-1}$，NaN$\odot x$ 等不明确的有限或无限的结果作任何运算均返回 NaN.

每当算术运算无效，由此产生 NaN 或上溢，或用零作除数产生 $\pm\infty$ 或下溢时，可设定一个异常标记，并且稍后可以用用户的程序测试. 这些特性使我们可以编写更可靠的程序(因为程序可以检测并校正它本身的异常，而不是简单地执行中断)和更快的程序(为了避免可能但未必出现的异常，应避免使用有许多检验指令和转移指令的“多疑的”程序设计). 例如问题 1.19、下面的引理 5.3 的注记和[81].

不适当地处理浮点异常会付出昂贵的代价，最惨痛的教训莫过于 1996 年 6 月 4 日欧洲航天局的阿里亚娜 5 火箭的坠落，其细节可见 HOME/ariane5rep. html.

显然几乎所有的机器都使用 IEEE 算术运算[1]做仔细的舍入，但不是所有的机器都这样. 比如由 Cray 公司[2]所生产的机器，虽然将来新的 Cray 机器也可能使用 IEEE 算术运算. 因为在一台 Cray 机器上计算的 $\mathrm{fl}(a\odot b)$ 和一台 IEEE 机器上计算的 $\mathrm{fl}(a\odot b)$ 之间的差别通常位于或超过第 14 个十进位，所以读者可能想知道差别是否重要. 实际上，数值线性代数中大多数算法对处理舍入方式的细节是不敏感的. 但是一旦正确地设定舍入后，我们会发觉一些算法是比较容易设计的，或是更可靠的. 下面是两个例子. [12]

当 Cray C90 从下一个较小的浮点数中减去 1 时，它得到 -2^{-47}，这是正确解 -2^{-48} 的两倍. 要保证求对称阵的特征值和特征向量的分而治之算法的正确性，得到一致的微小差别对相对高精度是重要的，可利用当前最快的算法处理这个问题. 为保证在 Cray 机器上的正确性，这个算法需要一个有点非直觉的修正(见 5.3.3 节).

在计算 $\arccos(x/\sqrt{x^2+y^2})$ 时，Cray 机器也可能得到一个误差，因为过分的舍入引起 arccos 的自变量大于 1. 这一点在 IEEE 算术运算中则不可能发生(见问题 1.17).

为了提供 Cray C90 或其他 Cray 机器上的误差分析，可以改为使用模型 $\mathrm{fl}(a\pm b)=a(1+\delta_1)\pm b(1+\delta_2)$，$\mathrm{fl}(a*b)=(a*b)(1+\delta_3)$，$\mathrm{fl}(a/b)=(a/b)(1+\delta_3)$，$|\delta_i|\leq\varepsilon$，其中 ε 是最大相对表示误差的一个小倍数.

简要地说，设计正确的舍入和其他 IEEE 算术运算的特性目的是尽可能地保持导

1. 包含诸如 NEC SX-4 这样的机器，这种机器有一种“Cray 模式”，它们以相同的方式执行算术运算. 把 DEC Alpha 处理机构造的并行计算机 Cray T3D 和 T3E 排除在外，这种机器十分接近于使用 IEEE 算术运算（由于速度的缘故上溢直接为零）.

2. Cray 公司在 1996 年被 SGI 公司兼并.

出公式中使用的许多数学关系. 若已知(上溢和下溢除外)$\mathrm{fl}(a-b)$是用小的相对误差计算的(相反分而治之法可能失败), 以及 $-1 \leq c \equiv \mathrm{fl}(x/\sqrt{x^2+y^2}) \leq 1$(相反 $\arccos(c)$ 可能失败), 则算法是容易设计的. 存在许多其他类似的依赖于(时常不知道的)设计算法的数学关系. 更多的关于 IEEE 算术运算的细节及其与数值分析的关系见[159,158,81].

已知浮点在遇到机器之后的可变性, 我们如何编写与运算有关的可移植的软件呢? 例如, 在后面章节中研究的迭代算法经常有下列循环

```
repeat
   ...
   update e
until   "与 f 相比 e 是可以忽略的"
```

[13]

其中 $e \geq 0$ 是某个误差测度, $f>0$ 是某个比较值(例见 4.4.5 节). "是 $e \leq c \cdot \varepsilon \cdot f$ 吗?" 我们意指可以忽略, 其中 $c \geq 1$ 是某个适中的常数, 用于在精确性和收敛速度之间取得平衡. 因为这个试验需要与机器相关的常数 ε, 过去这个试验通常用表面上与机器无关的试验 "是 $e+cf=cf$ 吗?" 来代替. 这里的想法是若 $e<c\varepsilon f$ 或许更小些, 则做 e 和 cf 加法, 再舍入从而得到 cf. 但是这个试验可能失败(要求 e 比必要的或者比可达到的数小很多), 它与所使用的机器及编译器有关(见下一段). 故最佳的试验当然明确地使用 ε. 结果是人们充分谨慎用一个与机器无关的方法计算 ε, 可以利用的软件是 LAPACK 中子程序 **slamch**(单精度)和 **dlamch**(双精度). 这些程序也计算或估计上溢阈值(而不上溢)、下溢阈值以及其他参数. 另一个使用这些明确的机器参数的可移植程序在问题 1.19 中讨论.

有时人们需要利用比 IEEE 单精度或双精度更高的精度. 例如, 诸如为了改进 $\boldsymbol{Ax}=\boldsymbol{b}$ 计算解精确度的迭代精化算法中用到较高的精度(见 2.5.1 节). 故 IEEE 定义另一个称为*双倍扩充*的更高的精度. 例如, 在一台英特尔奔腾处理器(或它的前身 8086/8087)上所有的算术运算在 80 位双倍扩充的寄存器内执行, 它提供 64 位尾数和 15 位指数. 遗憾地是, 不是所有的语言和编译器都允许人们用双倍扩充精度变量来显示和计算.

少量机器在硬件上提供除双倍扩充的运算之外的东西, 但有好几种方法用软件模拟更准确的运算. 在 DEC Vax 和 DEC Alpha, Sun Sparc 和 IBM RS6000 机器上的某些编译器允许用户显示*四倍精度*(或*实 * 16* 或*双倍双精度*)变量并对它们执行计算. 因为这个运算是利用较低的精度模拟的, 所以它可能比双精度运行慢好几倍. Cray 的单精度类似于 IEEE 双精度, 故 Cray 双精度大约是 IEEE 双精度的两倍. 它也是用软件进行模拟并且运行相对缓慢. 还有一些算法和软件包可以利用整数算术运算[20,21]或者利用优先的浮点指令(见问题 1.18)[204,218]来模拟更高精度浮点运算.

最后, 我们讨论*区间算术运算*, 这是一种自动提供有保证的误差界的计算类型.

在区间计算中，每个变量是用一对浮点数来表示的，一个是下界，一个是上界. 以一种有保证的方式传播下界和上界的方式作舍入进行计算. 例如，将区间 $a=[a_l,a_u]$ 和 $b=[b_l,b_u]$ 相加，向下舍入 a_l+b_l 到最接近的浮点数 c_l，向上舍入 a_u+b_u 到最接近的浮点数 c_u. 这样保证区间 $c=[c_l,c_u]$ 包括来自 a 中和来自 b 中的任何一对变量之和. 但是，如果人们天真地提出一个程序并把所有的浮点变量和运算转换成一个区间变量和运算，那么通过程序计算的区间极可能会迅速地变宽（像 $[-\infty,+\infty]$ 那样），它们提供根本无用的信息.（一个简单的例子是当 x 是一个区间时反复计算 $x=x-x$，每作一次减法，x 的宽度 x_u-x_l 加倍而不是得到 $x=0$.）修改老的算法或者设计提供实用的有保证的误差界的新算法[4、140、162、190]是有可能的，但是这些算法通常比本书中所讨论的算法多费时好几倍. 在区间界相同的数学意义下，不保证本书中提出的误差界是可靠的，但是它们在几乎所有情况下都是足够可靠的.（后面将更详尽地讨论这个问题.）在本书中我们将不讨论区间运算.

14

1.6 再议多项式求值

现在对用霍纳法则求多项式值应用舍入模型(1.1). 我们取原有的程序，

$p=a_d$
for $i=d-1$ down to 0
$\quad p=x\cdot p+a_i$
end for

然后对中间结果增加下标，使得对每个值有唯一的一个符号(p_0 是最后的结果)：

$p_d=a_d$
for $i=d-1$ down to 0
$\quad p_i=x\cdot p_{i+1}+a_i$
end for

然后对每个浮点运算插入一个舍入项$(1+\delta_i)$，得到：

$p_d=a_d$
for $i=d-1$ down to 0
$\quad p_i=((x\cdot p_{i+1})(1+\delta_i)+a_i)(1+\delta'_i)$, where $|\delta_i|,|\delta'_i|\leqslant\varepsilon$
end for

展开，我们得到下列多项式最后计算值的表达式：

$$p_0=\sum_{i=0}^{d-1}\left[(1+\delta'_i)\prod_{j=0}^{i-1}(1+\delta_j)(1+\delta'_j)\right]a_ix^i+\left[\prod_{j=0}^{d-1}(1+\delta_j)(1+\delta'_j)\right]a_dx^d.$$

15

当我们试图保留算法中每个舍入误差的踪迹时，这是一个杂乱的、有代表性的结果. 利用下列上界和下界来简化它.

$$(1+\delta_1)\cdots(1+\delta_j)\leqslant(1+\varepsilon)^j\leqslant\frac{1}{1-j\varepsilon}=1+j\varepsilon+O(\varepsilon^2),$$

$$(1+\delta_1)\cdots(1+\delta_j)\geqslant(1-\varepsilon)^j\geqslant 1-j\varepsilon.$$

倘若 $j\varepsilon<1$，这些界是正确的. 通常作合情合理的假定 $j\varepsilon\ll 1$（在 IEEE 单精度中 $j\ll 10^7$）并作近似

$$1-j\varepsilon\leqslant(1+\delta_1)\cdots(1+\delta_j)\leqslant 1+j\varepsilon.$$

由此，我们记

$$p_0=\sum_{i=0}^{d}(1+\bar{\delta}_i)a_ix^i=\sum_{i=0}^{d}\bar{a}_ix^i,\quad \text{其中}\ |\bar{\delta}_i|\leqslant 2d\varepsilon$$

故 $p(x)$ 的计算值 p_0 是具有系数 $\bar{a}_i$ 的一个稍许不同的多项式的精确值. 这意味着 $p(x)$ 求值是"向后稳定的"而度量 $p(x)$ 的任何系数最大相对改变的"向后误差"是 $2d\varepsilon$.

利用这个向后误差界，界定计算多项式中的误差为：

$$|p_0-p(x)|=\left|\sum_{i=0}^{d}(1+\bar{\delta}_i)a_ix^i-\sum_{i=0}^{d}a_ix^i\right|$$

$$=\left|\sum_{i=0}^{d}\bar{\delta}_ia_ix^i\right|\leqslant\sum_{i=0}^{d}\varepsilon 2d|a_i\cdot x^i|\leqslant 2d\varepsilon\sum_{i=0}^{d}|a_i\cdot x^i|.$$

注意，如果正数和负数相加过程中没有对消，则 $\sum_i|a_ix^i|$ 界定为我们可以计算的最大值并且误差界是 $2d\varepsilon$ 的较小倍数. 这也是计算点积和许多其他类似多项式表达式的情况.

选择 $\bar{\delta}_i=\varepsilon\cdot\text{sign}(a_ix^i)$，可以看到在适度的因子 $2d$ 范围内误差界是可达的. 这意味着可使用

16

$$\frac{\sum_{i=0}^{d}|a_ix^i|}{\left|\sum_{i=0}^{d}a_ix^i\right|}$$

作为多项式求值的相对条件数.

花费双倍运算量容易计算这个误差界：

```
p = a_d, bp = |a_d|
for i = d - 1 down to 0
    p = x · p + a_i
    bp = |x| · bp + |a_i|
end for
error bound = bp = 2d · ε · bp
```

故多项式的正确值位于区间 $[p-bp, p+bp]$ 之中，并且确保正确的十进制数字位数是 $-\log_{10}\left(\left|\frac{bp}{p}\right|\right)$. 对较早前讨论的多项式 $(x-2)^9$，这些界标示在图 1-3 的上部.（读者可能想知道舍入误差是否会使这个计算的误差界不准确. 这个证明不难，留给读者作为一个练习.）

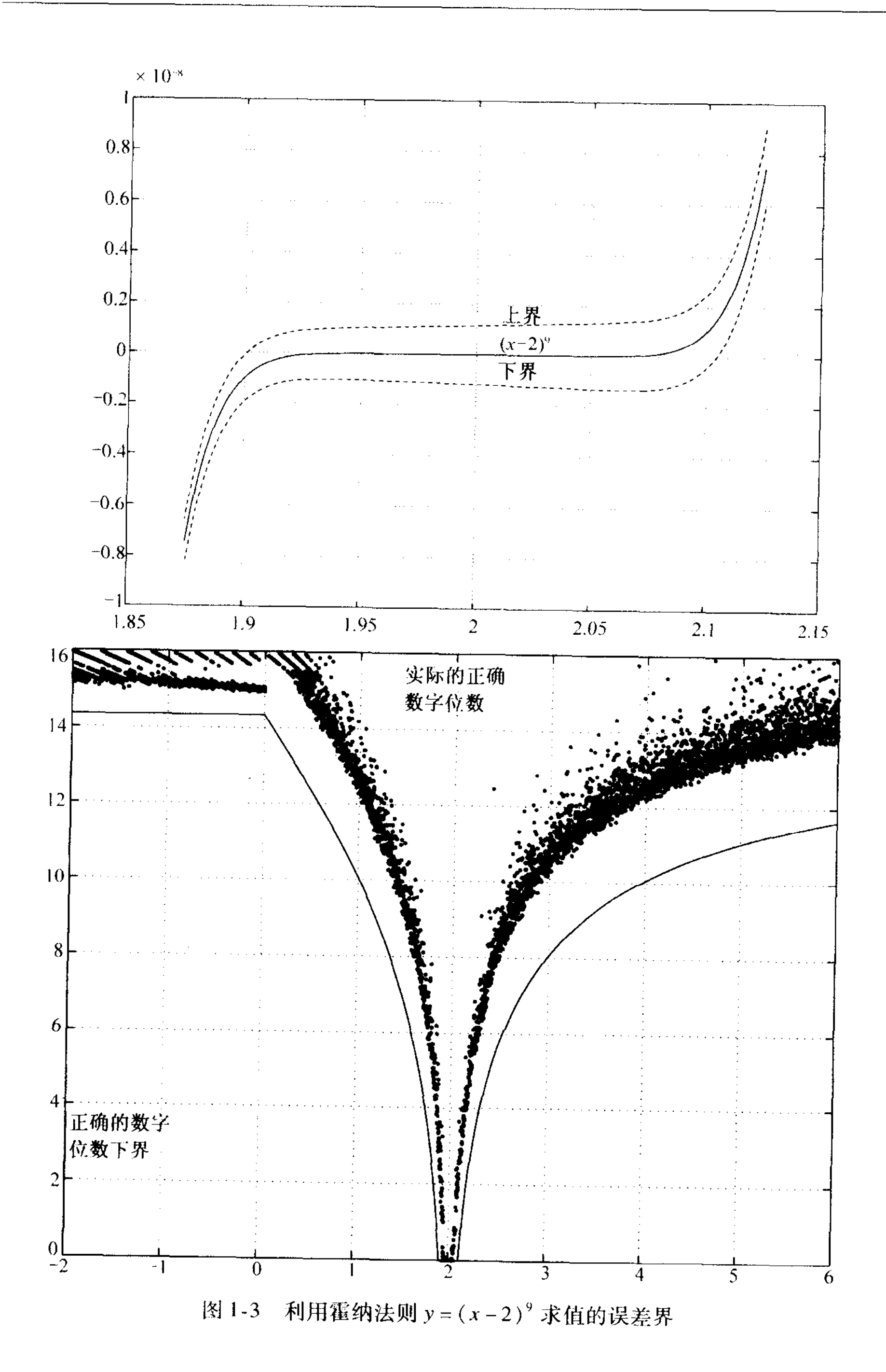


图 1-3 利用霍纳法则 $y=(x-2)^9$ 求值的误差界

在图1-3下部中关于正确的十进制数字位数的下界 $-\log_{10}\left|\frac{bp}{p}\right|$ 的图像指出当 $p(x)$ 接近于0时，计算 $p(x)$ 达到高的相对精度是相当困难的. 关于 $p(x)=0$ 的特别之处是什么呢？一个任意小的误差在计算 $p(x)=0$ 中引起一个无穷的相对误差 $\frac{\varepsilon}{p(x)}=\frac{\varepsilon}{0}$. 换言之，相对误差界 $2d\varepsilon\sum_{i=0}^{d}|a_ix^i|/\left|\sum_{i=0}^{d}a_ix^i\right|$ 无穷大.

定义1.1 条件数为无穷大的问题称为是不适定的，否则称为是适定的.[1]

条件数有一个简单的几何解释，它告诉我们 $p(x)$ 离一个不适定的多项式有多远.

定义1.2 设 $p(z)=\sum_{i=0}^{d}a_iz^i$ 和 $q(z)=\sum_{i=0}^{d}b_iz^i$. 定义 p 到 q 的相对距离 $d(p,q)$ 为满足 $|a_i-b_i|\leqslant d(p,q)\cdot|a_i|$ 的最小值，$0\leqslant i\leqslant d$.（若所有 $a_i\neq 0$，则我们可更简单地记作 $d(p,q)=\max_{0\leqslant i\leqslant d}\left|\frac{a_i-b_i}{a_i}\right|$.）

17 ~ 18

注意，若 $a_i=0$，则因 $d(p,q)$ 有限，b_i 必为零.

定理1.2 假如 $p(z)=\sum_{i=0}^{d}a_iz^i$ 不恒等于零，则

$$\min\{d(p,q):q(x)=0\}=\frac{\left|\sum_{i=0}^{d}a_ix^i\right|}{\sum_{i=0}^{d}|a_ix^i|}.$$

换言之，从 p 到最近的多项式 q 的距离是 $p(x)$ 的条件数的倒数，其中 q 的条件数在 x 处为无穷大，即 $q(x)=0$.

证明 记 $q(z)=\sum b_iz^i=\sum(1+\varepsilon_i)a_iz^i$ 使得 $d(p,q)=\max_i|\varepsilon_i|$. 则由 $q(x)=0$ 推出 $|p(x)|=|q(x)-p(x)|=\left|\sum_{i=0}^{d}\varepsilon_ia_ix^i\right|\leqslant\sum_{i=0}^{d}|\varepsilon_ia_ix^i|\leqslant\max_i|\varepsilon_i|\sum_i|a_ix^i|$，由此可得 $d(p,q)=\max|\varepsilon_i|\geqslant|p(x)|/\sum_i|a_ix^i|$. 为了明白存在一个靠近于 p 的 q，选择

$$\varepsilon_i=\frac{-p(x)}{\sum|a_ix^i|}\cdot\operatorname{sign}(a_ix^i).$$ □

这个条件数和到最近的不适定问题距离之间简单的倒数关系在数值分析中是非常普遍的，在后面将再次遇到它.

在本章的开头，我们说过为了帮助求解线性代数问题，将使用矩阵的典范型. 例如，知道精确的若尔当典范型使得计算精确的特征值微不足道. 对多项式而言，存在一个类似的典范型，它使得准确的多项式求值毫不费力：$p(x)=a_d\prod_{i=1}^{d}(x-r_i)$. 换言之，用它的首项系数 a_d 和它的根 $r_1,\cdots,r_n$ 来表示多项式. 为求 $p(x)$ 的值，

1. 这个定义稍稍有点非标准，因为不适定问题包含那些具有连续且不可微的解的问题. 例如多项式的重根以及矩阵的重特征值（4.3节）. 描述不适定问题的另一种方式是解中的正确数字位数不一定是求解的算术运算使用的固定的数字位数. 例如多项式重根失去一半或更多的算术运算精度.

我们利用显而易见的算法

```
p = a_d
for i = 1 to d
    p = p · (x - r_i)
end for
```

容易证明计算的 $p=p(x)\cdot(1+\delta)$，其中 $|\delta|\leqslant 2d\varepsilon$. 即总能得到具有高的相对精度的 $p(x)$. 但是要做到这一点我们需要多项式的根.

1.7 向量和矩阵范数

范数（norm）用于度量矩阵计算中的误差，故需要理解如何计算和熟练地操作它们.

证明留作本章的问题.

定义 1.3 设$\mathcal{B}$是一个实(复)线性空间$\mathbb{R}^n$(或$\mathbb{C}^n$). 这个空间是赋范的，若存在一个函数$\|\cdot\|$: $\mathcal{B}\to\mathbb{R}$，它满足下列关系：

19

1) $\|\boldsymbol{x}\|\geqslant 0$，且$\|\boldsymbol{x}\|=0$ 当且仅当 $\boldsymbol{x}=0$(正定性)；

2) $\|\alpha\boldsymbol{x}\|=|\alpha|\cdot\|\boldsymbol{x}\|$，对一切实(或复)标量 α 成立(齐次性)；

3) $\|\boldsymbol{x}+\boldsymbol{y}\|\leqslant\|\boldsymbol{x}\|+\|\boldsymbol{y}\|$(三角形不等式).

我们称此函数为范数.

例 1.4 最常用的范数是$\|\boldsymbol{x}\|_p=(\sum_i|x_i|^p)^{1/p}$，$1\leqslant p<\infty$，称之为$p$-范数，又$\|\boldsymbol{x}\|_\infty=\max_i|x_i|$，称之为$\infty$-范数或无穷范数. 此外，若$\|\boldsymbol{x}\|$是任意的范数而$\boldsymbol{C}$是任意的非异矩阵，则$\|\boldsymbol{C}\boldsymbol{x}\|$也是一个范数. ◇

有许多可以用于度量误差的范数，重要的是选择一个适当的范数. 例如，设 $\boldsymbol{x}_1=[1,2,3]^{\mathrm{T}}$，分量单位是米，$\boldsymbol{x}_2=[1.01,2.01,2.99]^{\mathrm{T}}$，分量单位是米，则$\boldsymbol{x}_2$ 是$\boldsymbol{x}_1$ 的一个好的近似，因为相对误差$\frac{\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_\infty}{\|\boldsymbol{x}_1\|_\infty}\approx 0.0033$，而 $\boldsymbol{x}_3=[10,2.01,2.99]^{\mathrm{T}}$ 是一个不好的近似，因为$\frac{\|\boldsymbol{x}_1-\boldsymbol{x}_3\|_\infty}{\|\boldsymbol{x}_1\|_\infty}=3$. 但是假定第一个分量用千米而不是米来度量，则在这个范数下 $\hat{\boldsymbol{x}}_1$ 和 $\hat{\boldsymbol{x}}_3$ 比较接近.

$$\hat{\boldsymbol{x}}_1=\begin{bmatrix}0.001\\2\\3\end{bmatrix},\ \hat{\boldsymbol{x}}_3=\begin{bmatrix}0.01\\2.01\\2.99\end{bmatrix},\ \text{且}\frac{\|\hat{\boldsymbol{x}}_1-\hat{\boldsymbol{x}}_3\|_\infty}{\|\hat{\boldsymbol{x}}_1\|_\infty}\approx 0.0033.$$

为了比较 $\hat{\boldsymbol{x}}_1$ 和 $\hat{\boldsymbol{x}}_3$，应该使用

$$\|\hat{\boldsymbol{x}}\|_s=\left\|\begin{bmatrix}1000&&\\&1&\\&&1\end{bmatrix}\hat{\boldsymbol{x}}\right\|_\infty$$

达到单位相同或者使重要的误差获得同样大的范数.

现在定义内积，它是标准点积 $\sum_i x_i y_i$ 的推广，而且其在线性代数中经常出现.

定义 1.4 设$\mathcal{B}$是一个实(复)线性空间. $\langle\cdot,\cdot\rangle$: $\mathcal{B}\times\mathcal{B}\to\mathbb{R}$ ($\mathbb{C}$)是一个内积，如果它适合下面所有条件：

1) $\langle x,y\rangle=\langle y,x\rangle$(或$\overline{\langle y,x\rangle}$)；

2) $\langle x,y+z\rangle=\langle x,y\rangle+\langle x,z\rangle$；

3) $\langle\alpha x,y\rangle=\alpha\langle x,y\rangle$，对一切实(或复)标量$\alpha$；

4) $\langle x,x\rangle\geqslant 0$，当且仅当$x=0$时等号成立.

例 1.5 在$\mathbb{R}$上，$\langle x,y\rangle=y^{\mathrm{T}}x=\sum_i x_i y_i$，而在$\mathbb{C}$上，$\langle x,y\rangle=y^*x=\sum_i x_i\bar{y}_i$ 是内积.（记住$y^*=\bar{y}^{\mathrm{T}}$是y的共轭转置） ◇

[20] **定义 1.5** 若$\langle x,y\rangle=0$，则称x和y是正交的.

内积最重要的性质是它满足柯西-施瓦茨不等式. 此式能用于证明$\sqrt{\langle x,x\rangle}$是一个范数，我们将经常使用这个范数.

引理 1.1 柯西-施瓦茨不等式. $|\langle x,y\rangle|\leqslant\sqrt{\langle x,x\rangle\cdot\langle y,y\rangle}$.

引理 1.2 $\sqrt{\langle x,x\rangle}$是一个范数.

内积和对称(埃尔米特)正定阵之间存在一个如下定义的1-1对应，这些矩阵在应用中经常用到.

定义 1.6 若对一切$x\neq 0$，有$x^{\mathrm{T}}Ax>0$($x^*Ax>0$)成立，则称实对称(复埃尔米特)阵A是正定的. 简写对称正定为s.p.d，埃尔米特正定为h.p.d.

引理 1.3 设$\mathcal{B}=\mathbb{R}^n$(或$\mathbb{C}^n$)，$\langle\cdot,\cdot\rangle$为一个内积. 则存在一个$n\times n$阶s.p.d(h.p.d)阵$\boldsymbol{A}$使得$\langle x,y\rangle=y^{\mathrm{T}}\boldsymbol{A}x$($y^*\boldsymbol{A}x$). 反之，若$\boldsymbol{A}$是s.p.d(h.p.d)，则$y^{\mathrm{T}}\boldsymbol{A}x$($y^*\boldsymbol{A}x$)是一个内积.

一种范数形式的误差界转换成另一种范数形式的误差界时，下列两个引理是有用的.

引理 1.4 设$\|\cdot\|_\alpha$和$\|\cdot\|_\beta$是$\mathbb{R}^n$(或$\mathbb{C}^n$)上的两个范数，存在常数c_1，$c_2>0$使得对于一切x，$c_1\|x\|_\alpha\leqslant\|x\|_\beta\leqslant c_2\|x\|_\alpha$. 我们也称范数$\|\cdot\|_\alpha$和$\|\cdot\|_\beta$关于常数$c_1$和$c_2$等价.

引理 1.5

$$\|\boldsymbol{x}\|_2\leqslant\|\boldsymbol{x}\|_1\leqslant\sqrt{n}\|\boldsymbol{x}\|_2,$$
$$\|\boldsymbol{x}\|_\infty\leqslant\|\boldsymbol{x}\|_2\leqslant\sqrt{n}\|\boldsymbol{x}\|_\infty,$$
$$\|\boldsymbol{x}\|_\infty\leqslant\|\boldsymbol{x}\|_1\leqslant n\|\boldsymbol{x}\|_\infty.$$

除向量范数之外，我们也需要度量矩阵中误差的矩阵范数.

定义 1.7 $\|\cdot\|$是关于$m\times n$阶矩阵的一个矩阵范数，如果它是$m\cdot n$维空间上的一个向量范数：

1) $\|\boldsymbol{A}\|\geqslant 0$，当且仅当$\boldsymbol{A}=0$时等号成立；

2) $\|\alpha A\| = \|\alpha\| \cdot \|A\|$;

3) $\|A + B\| \leqslant \|A\| + \|B\|$.

例 1.6 $\max_{ij} |a_{ij}|$称为最大范数，而$(\sum |a_{ij}|^2)^{1/2} = \|A\|_F$称为弗罗贝尼乌斯(Frobenius)范数. ◇

21

当导出误差界时通常需要界定矩阵乘积的范数，此时，下列定义是有用的.

定义 1.8 设$\|\cdot\|_{m\times n}$是关于$m\times n$阶矩阵的一个矩阵范数，$\|\cdot\|_{n\times p}$是关于$n\times p$阶矩阵的一个矩阵范数，而$\|\cdot\|_{m\times p}$是关于$m\times p$阶矩阵的一个矩阵范数. 如果$\|A\cdot B\|_{m\times p} \leqslant \|A\|_{m\times n}\cdot\|B\|_{n\times p}$，其中$A$是$m\times n$阶矩阵，$B$是$n\times p$阶矩阵，则这些范数称为相互相容的.

定义 1.9 设A是$\hat{m}\times\hat{n}$阶矩阵，$\|\cdot\|_{\hat{m}}$是$\mathbb{R}^m$上的一个向量范数，$\|\cdot\|_{\hat{n}}$是$\mathbb{R}^n$上的一个向量范数，则

$$\|A\|_{\hat{m}\hat{n}} \equiv \max_{\substack{x\neq 0\\ x\in\mathbb{R}^n}} \frac{\|Ax\|_{\hat{m}}}{\|x\|_{\hat{n}}}$$

称为一个算子范数，或导出范数，或从属矩阵范数.

下列引理提供大量的矩阵范数源，为了界定误差，我们将使用下列范数.

引理 1.6 算子范数是一个矩阵范数.

对于几乎所有最小二乘问题和特征值问题的算法来说，下面定义的正交阵和酉阵是基本的组成部分.

定义 1.10 若$Q^{-1} = Q^{\mathrm{T}}$，称实方阵Q是正交阵，若$Q^{-1} = Q^*$，则称复方阵Q是酉阵.

正交(或酉阵)的所有行(或列)有单位 2-范数，并且因为$QQ^{\mathrm{T}} = Q^{\mathrm{T}}Q = I(QQ^* = Q^*Q = I)$，所以它们彼此是正交的.

下列引理总结了迄今为止引进的范数和矩阵的基本性质. 在本书的后面将使用这些性质.

引理 1.7 1. $\|Ax\| \leqslant \|A\|\cdot\|x\|$，对一个向量范数及其对应的算子范数，或向量 2-范数和矩阵弗罗贝尼乌斯范数成立.

2. $\|AB\| \leqslant \|A\|\cdot\|B\|$，对任何算子范数或弗罗贝尼乌斯范数成立. 换言之，任何算子范数(或弗罗贝尼乌斯范数)和它本身是相互相容的.

3. 最大范数和弗罗贝尼乌斯范数不是算子范数.

4. $\|QAZ\| = \|A\|$，当Q和Z是正交阵或酉阵时，对弗罗贝尼乌斯范数及由$\|\cdot\|_2$导出的算子范数成立. 而这实际上正好是毕达哥拉斯定理.

22

5. $\|A\|_\infty \equiv \max_{x\neq 0}\frac{\|Ax\|_\infty}{\|x\|_\infty} = \max_i \sum_j |a_{ij}| =$ 最大的绝对值行和.

6. $\|A\|_1 \equiv \max_{x\neq 0}\frac{\|Ax\|_1}{\|x\|_1} = \|A^{\mathrm{T}}\|_\infty = \max_j \sum_i |a_{ij}| =$ 最大的绝对值列和.

7. $\|A\|_2 \equiv \max_{x\neq 0}\frac{\|Ax\|_2}{\|x\|_2} = \sqrt{\lambda_{\max}(A^*A)}$，其中$\lambda_{\max}$表示最大的特征值.

8. $\|A\|_2 = \|A^T\|_2$.

9. $\|A\|_2 = \max_i |\lambda_i(A)|$，当 A 是正规阵，即 $AA^* = A^*A$ 时成立.

10. 若 A 是 $n \times n$ 阶矩阵，则 $n^{-1/2}\|A\|_2 \leq \|A\|_1 \leq n^{1/2}\|A\|_2$.

11. 若 A 是 $n \times n$ 阶矩阵，则 $n^{-1/2}\|A\|_2 \leq \|A\|_\infty \leq n^{1/2}\|A\|_2$.

12. 若 A 是 $n \times n$ 阶矩阵，则 $n^{-1}\|A\|_\infty \leq \|A\|_1 \leq n\|A\|_\infty$.

13. 若 A 是 $n \times n$ 阶矩阵，则 $\|A\|_1 \leq \|A\|_F \leq n^{1/2}\|A\|_2$.

证明 我们只证明第7部分，其余当作问题1.16.

因为 A^*A 是埃尔米特矩阵，所以存在一个特征分解 $A^*A = Q\Lambda Q^*$，其中，Q 是酉阵(列是特征向量)，$\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ 是包含特征值的对角阵，而特征值必定都是实的. 注意所有的 $\lambda_i \geq 0$，因为若某个 λ 是负的，则我们取 q 为它的特征向量，有 $0 \leq \|Aq\|_2^2 = q^T A^T A q = q^T \lambda q = \lambda \|q\|_2^2 < 0$，矛盾. 所以

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \max_{x \neq 0} \frac{(x^* A^* A x)^{1/2}}{\|x\|_2} = \max_{x \neq 0} \frac{(x^* Q \Lambda Q^* x)^{1/2}}{\|x\|_2}$$

$$= \max_{x \neq 0} \frac{((Q^* x)^* \Lambda Q^* x)^{1/2}}{\|Q^* x\|_2} = \max_{y \neq 0} \frac{(y * \Lambda y)^{1/2}}{\|y\|_2} = \max_{y \neq 0} \frac{\sqrt{\sum \lambda_i y_i^2}}{\sqrt{\sum y_i^2}}$$

$$\leq \max_{y \neq 0} \sqrt{\lambda_{\max}} \sqrt{\frac{\sum y_i^2}{\sum y_i^2}} = \sqrt{\lambda_{\max}},$$

当选择 y 是单位阵的适当的列时，上界是可以达到的. □

1.8 第1章的参考书目和其他话题

在每一章的结尾，我们将列举许多与该章有关的参考书目. 它们也按字母顺序列在最后的参考文献中. 此外，对教材正文中未讨论的有关话题也将给出指示性的信息.

在这个领域中，G. Golub and C. Van Loan[121]是内容最为丰富的一本书，该书中附有大量的参考文献. 最近出版的 D. Watkins[252]是大学本科生或研究生低年级的教材. 另一本优秀的研究生教材是 L. Trefethen and D. Bau [243]. J. Wilkinson [262]虽出版较早但仍很经典，G. Stewart[235]出版较早，但仍是与 Watkins 的书水平相当的优秀著作.

23

关于误差分析更加详尽的信息可以在 N. Higham 最新的著作[149]中找到. J. Wilkinson[261]和 W. Kahan[157]虽然出版时间较早，但仍不失为一本很好的参考书.

D. Goldberg 的论文 *What every computer scientist should know about floating point arithmetic* 是较新的一篇优秀综述文章[119]. IEEE 算术运算在[11,12,159]中以及计算机厂商发行的参考手册中有正式的论述. 关于 IEEE 算术运算误差分析的讨论可以在[54,70,159,158]及其引用的参考书目中找到.

有关条件数以及与最接近不适定问题的距离的更一般的讨论，是由笔者在[71]中以及 S. Smale 和 M. Shub[219,220,221,222]的系列论文中给出的. 向量和矩阵范数在[121 的 2.2，2.3 节]中作了详尽的讨论.

1.9 第1章问题

问题 1.1(容易. Z. Bai) 设 $\boldsymbol{A}$ 是一个正交阵，证明 $\det(\boldsymbol{A})=\pm 1$. 证明若 $\boldsymbol{B}$ 也是正交的且 $\det(\boldsymbol{A})=-\det(\boldsymbol{B})$，则 $\boldsymbol{A}+\boldsymbol{B}$ 奇异.

问题 1.2(容易. Z. Bai) 矩阵的秩是它的列张成空间的维数. 证明：$\boldsymbol{A}$ 的秩为 1 当且仅当对某些列向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$，$\boldsymbol{A}=\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}$.

问题 1.3(容易. Z. Bai) 证明：若矩阵是正交的和三角形的，则它是对角阵. 它的对角元是什么?

问题 1.4(容易. Z. Bai) 矩阵是严格上三角形的，如果它是具有零对角元的上三角形阵. 证明：若 $\boldsymbol{A}$ 是 $n\times n$ 阶严格上三角阵，则 $\boldsymbol{A}^n=0$.

问题 1.5(容易. Z. Bai) 设$\|\cdot\|$是$\mathbb{R}^m$ 上的一个向量范数，并假定 $\boldsymbol{C}\in\mathbb{R}^{m\times n}$. 证明：若 $\operatorname{rank}(\boldsymbol{C})=n$，则$\|x\|_C\equiv\|\boldsymbol{C}x\|$是一个向量范数.

问题 1.6(容易. Z. Bai) 证明：若 $0\neq \boldsymbol{s}\in\mathbb{R}^n$，$\boldsymbol{E}\in\mathbb{R}^{n\times n}$，则

$$\left\|\boldsymbol{E}\left(\boldsymbol{I}-\frac{\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}}}{\boldsymbol{s}^{\mathrm{T}}\boldsymbol{s}}\right)\right\|_F^2=\|\boldsymbol{E}\|_F^2-\frac{\|\boldsymbol{E}\boldsymbol{s}\|_2^2}{\boldsymbol{s}^{\mathrm{T}}\boldsymbol{s}}.$$

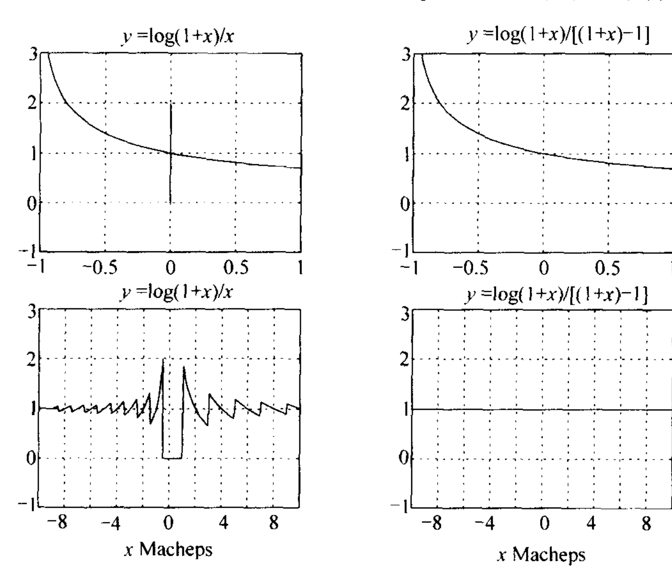
问题 1.7(容易. Z. Bai) 对任意的 $\boldsymbol{x}$，$\boldsymbol{y}\in\mathbb{C}^n$，验证$\|\boldsymbol{x}\boldsymbol{y}^{\mathrm{H}}\|_F=\|\boldsymbol{x}\boldsymbol{y}^{\mathrm{H}}\|_2=\|\boldsymbol{x}\|_2\|\boldsymbol{y}\|_2$. 24

问题 1.8(中等) 可以把 d 次多项式 $p(x)=\sum_{i=0}^{d}a_ix^i$ 和由系数向量构成的$\mathbb{R}^{d+1}$ 等同起来. 设 x 为定值. 设 S_x 为多项式集，它们在 x 处求值具有无限相对条件数(即它们在 x 处为0). 在几何上简单描述 S_x 为$\mathbb{R}^{d+1}$的一个子集. 设 $S_x(\kappa)$是相对条件数为 κ 或更大一些的多项式集. 在几何上简单描述 $S_x(\kappa)$. 描述当 $\kappa\to\infty$ 时$S_x(\kappa)$在几何上怎样变化.

问题 1.9(中等) 考虑下列图形. 它描绘用两种不同的方法计算函数 $y=\log(1+x)/x$的图像. 数学上，在靠近 $x=0$ 处 y 是 x 的一个光滑函数，在 $x=0$ 处 $y=1$. 但是如果利用这个公式计算 y，则我们得到左边的图像（左上部表示$x\in[-1,1]$的范围，左下部表示 $x\in[-10^{-15},10^{-15}]$的范围). 在靠近 $x=0$ 处这个公式显然不稳定. 另一方面，若利用下列算法

```
d = 1 + x
if d = 1 then
    y = 1
else
    y = log(d)/(d - 1)
end if
```

我们得到右边两个图像，它们在靠近 $x=0$ 处是正确的. 解释这个现象，证明按浮点算术运算第二个算法肯定得到一个精确的解. 假定 log 函数对任何自变量都获得精确的解.（这对对数的任意合理的实现过程都是成立的.）假定如此，自变量 IEEE 浮点算术运算将比较容易.（两个算法同时在一台 Cray 机器上可能发生故障.）



25

问题 1.10（中等）　不包括上溢和下溢，证明 $\mathrm{fl}(\sum_{i=1}^{d}x_iy_i)=\sum_{i=1}^{d}x_iy_i(1+\delta_i)$，其中 $|\delta_i|\leqslant d\varepsilon$. 利用此式证明下列事实. 设 $\boldsymbol{A}^{m\times n}$ 和 $\boldsymbol{B}^{n\times p}$ 是矩阵，用通常的方法计算它们的积. 不包括上溢和下溢，证明 $|\mathrm{fl}(\boldsymbol{A}\cdot\boldsymbol{B})-\boldsymbol{A}\cdot\boldsymbol{B}|\leqslant n\cdot\varepsilon\cdot|\boldsymbol{A}|\cdot|\boldsymbol{B}|$，这里矩阵的绝对值 $|\boldsymbol{A}|$ 指具有元素 $(|\boldsymbol{A}|)_{ij}=|a_{ij}|$ 的矩阵，并且该不等式按分量有意义.

该问题的结论将在 2.4.2 节中用于分析高斯消元法中的舍入误差.

问题 1.11（中等）　设 $\boldsymbol{L}$ 是下三角阵，用向前回代求解 $\boldsymbol{L}x=b$. 证明：不包括上溢和下溢，求得的解 $\hat{x}$ 满足 $(\boldsymbol{L}+\delta\boldsymbol{L})\hat{x}=b$，其中 $|\delta l_{ij}|\leqslant n\varepsilon|l_{ij}|$，$\varepsilon$ 是机器精度. 这意味着向前回代是向后稳定的. 证明解上三角形线性方程组的向后回代满足同样的界.

这个问题的结论将在 2.4.2 节中用于分析高斯消元法中的舍入误差.

问题 1.12（中等）　为了分析舍入误差的影响，使用下列模型[见(1.1)式]：

$$\mathrm{fl}(a\odot b)=(a\odot b)(1+\delta),$$

其中 $\odot$ 是四种基本运算 $+$，$-$，$*$ 和 $/$ 之一，且 $|\delta|\leqslant\varepsilon$. 为证明我们的分析对复数据也起作用，对四种基本的复运算需要证明一个类似的公式. 设 δ 是一个绝对值以 ε 的小倍数为界的微小的复数. 证明：这个公式对复数的加法、减法、乘法和除法是成立的. 对复数除法的算法应该能够成功地计算 $a/a\approx1$，其中 $|a|$ 或者是非常大（大于上溢阈值的平方根）或者是非常小（小于下溢阈值的平方根）. 复数乘积中的实部和虚部都能够达到高的相对精度吗？

问题1.13(中等) 证明引理1.3.

问题1.14(中等) 证明引理1.5.

问题1.15(中等) 证明引理1.6.

问题1.16(中等) 证明引理1.7中除第7部分之外的所有结论. 第8部分提示：利用若X和Y都是$n\times n$阶矩阵，则XY和YX有相同的特征值. 第9部分提示：利用矩阵是正规的当且仅当它有一个完全的规范正交特征向量组.

26

问题1.17(困难. W. Kahan) 我们曾提及在一台Cray机上因为$(x/\sqrt{x^2+y^2})$引起的舍入误差超过1，而使表达式$\arccos(x/\sqrt{x^2+y^2})$发生一个误差. 证明若不包括上溢和下溢，利用IEEE算术运算得不到该结果. 提示：需要利用比下列简单模型更多的东西：$\mathrm{fl}(a\odot b)=(a\odot b)(1+\delta)$，$|\delta|$很小. 考虑$\sqrt{x^2}$求值，不包括上溢和下溢，证明$\mathrm{fl}(\sqrt{x^2})$正好等于$x$. 数值实验由A. Liu做的，在Cray YMP上大约有5%的次数实验失败. 你可以尝试某个数值实验并说明它们. 进一步证明下述结论可有额外的学分：利用正确地舍入的十进制算术运算证明同样的结果.（证明是不同的）这个问题是W. Kahan受J. Sethian的Cray程序中一个程序缺陷的启发后给出的.

问题1.18(困难) 假如a和b是规格化的IEEE双精度浮点数，考虑下列用IEEE算术运算运行的算法：

if $(|a|<|b|)$, swap a and b

$s_1=a+b$

$s_2=(a-s_1)+b$

证明下列事实：

1. 不包括上溢和下溢，在运行计算$s_1=\mathrm{fl}(a+b)$的算法过程中只做舍入误差. 换言之，减法s_1-a和$(s_1-a)-b$两者都被精确地算出.

2. $s_1+s_2=a+b$，精确地成立. 这意味着s_2实际上是当舍入$a+b$的精确值得到s_1时所做的舍入误差.

从而，这个程序实际上模拟4倍精度算术运算，把真正的和$a+b$表示为高阶位(s_1)与低阶位(s_2)之和.

在一个系统的方法中利用这一点以及类似的窍门，就可以只利用优先的无"伪造位"(bitfiddling)[204]的浮点指令来达到以任意精度算术运算来有效地模拟所有四种基本的浮点运算. 这个方法在IBM RS6000和Cray上实现了128位运算.（但是在Cray上效率不及前者，因为Cray上没有IEEE算术运算.）

问题1.19(困难. 程序设计) 这个问题说明了开发高可靠性数值软件所面临的挑战. 给定$x_1,\cdots,x_n$. 要求编写计算2-范数$s\equiv\|\boldsymbol{x}\|_2=(\sum_{i=1}^{n}x_i^2)^{1/2}$的程序. 最明显的(但是不适当的)算法是

$s=0$

for $i=1$ to n

[27]

$s = s + x_i^2$

end for

$s = \text{sqrt}(s)$

这个算法是不适当的，因为它不具有下面所述的性质：

1. 它必须精确地计算解(即几乎所有计算的数位必须是正确的)，除非$\|x\|_2$(几乎)超过规格化浮点的取值范围.

2. 在大多数情况下它的运行速度必须几乎与上面的明显程序一样快.

3. 它必须在任何“正常的”机器，其中可能包括不运行 IEEE 算术运算的机器上工作. 这意味着不可以引入舍入误差，除非$\|x\|_2$(几乎)大于最大的浮点数.

要说明其中存在的困难，可以注意到当 $n=1$ 以及 x_1 大于最大的浮点数的平方根时(此时 x_1^2 上溢，在 IEEE 算术运算中程序获得 $+\infty$ 而在大多数非 IEEE 算术运算中程序中止)，或者当 $n=1$ 以及 x_1 小于最小的规格化浮点数的平方根时(此时 x_1^2 下溢，可能趋向零，算法可能获得零)，这一明显算法将失效. 用 $\max_i |x_i|$来除所有的 x_i，按比例缩小 x_i 不具有性质 2. 因为除法通常比乘法或者加法费时多得多. 即使 $\max_i |x_i|>0$，用 $c=1/\max_i|x_i|$去乘要冒计算 c 时上溢的危险.

这个程序是相当重要的，它已被标准化为 BLAS 的一个子程序，BLAS 可以在各种机器上使用[169]. 在 2.6.1 节我们将详细讨论 BLAS，文档和范例实现可以在 NETLIB/blas 上找到. 特别地，有性质 1 和 3 但没有性质 2 的范例实现可见 NETLIB/cgi-bin/netlibget.pl/blas/snrm2.f. 这些范例实现旨在成为针对专门计算机体系结构而设计的实现的起点. (这与本题所要求的完全可移植的程序相比更容易). 从而，当编写你自己的数值软件时，应该考虑把计算$\|\boldsymbol{x}\|_2$ 作为每台机器的数值库中可利用的一个模块.

$\|\boldsymbol{x}\|_2$ 的另一个精心的实现见[35].

你可以从 NETLIB/blas/sblat1 中选取试验代码来检验你的实现方法是否正确. 所有的实现必须彻底地测试并计时，而且要与上面的明显算法比较时间. 看看如何接近满足你能够达到的三个条件. 在条件 (1)、(2) 和 (3) 中经常使用“几乎”这样的词，它表示为了能够满足另一条件，在满足一个条件时，可能需要折中一下. 特别地，你可能希望更多了解若能够运行 IEEE 算术运算，能使问题有多大程度的简化.

[28]

提示：假定你的算法可利用上溢阈值和下溢阈值. 可以利用计算这些值的可移植软件(见 NETLIB/cgi-bin/netlibget.pl/lapack/util/slamch.f).

问题 1.20(容易. 中等) 我们将用 Matlab 程序说明多项式的根对系数的小扰动有多么敏感. 程序在 HOMEPAGE/Matlab/polyplot.m 上可以找到[1]. Polyplot 取一个由多项式的 r 个根所确定的输入多项式，然后加随机扰动到多项式的系数上，计算扰动的根并在图上标出它们的位置. 输入是

1. 请记住，在教材中把课程主页 URL 地址简写为 HOMEPAGE.

r = 多项式的根向量.

e = 多项式的每个系数所获得的最大相对扰动.

m = 生成在图上标出多项式根的随机多项式的个数，多项式的根已在图上标出.

1.(容易)任务的第一部分是对下面的输入运行这个程序. 在所有情况下，选择 m 足够大，得到一个相当稠密的图表，但不必等待太久. m = 几百或许1000就足够了. 如果图像太小或者太大，你可能要改变图表的轴.

- $r=(1:10)$;$e=1e-3,1e-4,1e-5,1e-6,1e-7,1e-8$.
- $r=(1:20)$;$e=1e-9,1e-11,1e-13,1e-15$.
- $r=[2,4,8,16,\cdots,1024]$;$e=1e-1,1e-2,1e-3,1e-4$.

你也可以尝试自己的具有复共轭根的例子，观察哪些根最敏感呢?

2.(中等)任务的第二部分是修改程序，对每个根修正计算条件数c(i). 换言之，在每个系数中的相对扰动e，至多用约e * c(i)改变根r(i). 修正程序画出中心在r(i)半径为e * c(i)的圆，并确认这些圆围住扰动的根(至少当e足够小用来导出条件数的线性化是准确的). 你应该完成几个有圆和扰动特征值的图像，并对观察到的情况作一些说明.

3.(中等)最后部分，注意若 $p'(\mathrm{r(i)})=0$，则c(i)的公式"放大". 这个条件意味着r(i)是 $p(x)=0$ 的重根. 我们仍可期望重根的计算值中的某种精确度，然而，在问题的这个部分，我们将要问重根敏感程度会怎样? 首先，记 $p(x)=q(x)\cdot(x-\mathrm{r(i)})^m$ 其中 $q(\mathrm{r(i)})\neq 0$，m 是根r(i)的重数. 然后计算小扰动多项式 $p(x)-q(x)\varepsilon$ 的最接近于r(i)的 m 个根，证明它们与r(i)的差为 $|\varepsilon|^{1/m}$，例如当 $m=2$ 时. 根r(i)被扰动 $\varepsilon^{1/2}$，当 $|\varepsilon|\ll 1$ 时，$\varepsilon^{1/2}$ 比 ε 大得多. 较大的解确实得到较大的扰动. 若 ε 约为机器精度并表示计算根的舍入误差，这意味着 m 重根可能丢失全部有效数字而不是它的有效数字位数的 $1/m$. [29]

问题1.21(中等) 应用对分算法1.1求 $p(x)=(x-2)^9=0$ 的根，其中 $p(x)$ 利用霍纳法则求值. 利用HOMEPAGE/Matlab/bisect.m中的Matlab执行，或者编写你自己的程序. 证明输入区间微小的改变会导致计算根巨大的改变. 当得到的 $p(x)$ 的计算值中的舍入误差变得非常大以致它的符号无法确定时停止二分法，利用本书中讨论的误差界修改算法. [30]

第2章 线性方程组求解

2.1 概 述

本章讨论线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 求解的扰动理论、算法和误差分析. 算法都是高斯消元法的变形. 它们称为*直接法*，因为若没有舍入误差，则有限步之后它们将给出 $\boldsymbol{Ax}=\boldsymbol{b}$ 的精确解. 与直接法相对，第6章讨论的*迭代法*是计算不断地更好逼近 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解的序列 $x_0,x_1,x_2,\cdots$，当 x_i 足够精确时我们停止迭代(计算下一个 x_{i+1}). 直接法或者迭代法可能更快些或者更精确些，这与矩阵 $\boldsymbol{A}$ 以及与 x_i 收敛于 $\boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{b}$ 的速度有关. 在第6章最后将讨论直接法和迭代法的优点. 到目前为止，我们只能说，当用户没有获得有关矩阵 $\boldsymbol{A}$ 来源[1]的特别知识或者一个解所需满足的稳定性和时间总量时，直接法是可选的方法.

本章的其余部分组织如下：2.2节讨论 $\boldsymbol{Ax}=\boldsymbol{b}$ 的扰动理论，它构成2.4节中实际误差界的基础；2.3节导出稠密矩阵的高斯消元法；2.4节分析高斯消元法中的误差并给出实际的误差界；2.5节指出如何利用一个简单和花费不多的迭代法来改进高斯消元法解的精度；为了在现代的计算机上得到高斯消元法和其他线性代数算法的高速度，必须注意统筹计算以更好地考虑计算机存储器结构，这个内容在2.6节中讨论；最后，在2.7节中讨论通常发生在实际问题中具有特殊性质矩阵的高斯消元法，诸如具有对称性质($\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$)或者稀疏性质(当 $\boldsymbol{A}$ 的许多元素为零时)的矩阵.

2.2.1节和2.5.1节讨论基于LAPACK库中软件的最新方法.

此外目前还有各种未解决的问题，将在下文中逐步指出.

2.2 扰动理论

假定 $\boldsymbol{Ax}=\boldsymbol{b}$ 和 $(\boldsymbol{A}+\delta\boldsymbol{A})\hat{\boldsymbol{x}}=\boldsymbol{b}+\delta\boldsymbol{b}$；我们的目标是界定 $\delta\boldsymbol{x}\equiv\hat{\boldsymbol{x}}-\boldsymbol{x}$ 的范数. 以后，$\hat{x}$ 将是 $\boldsymbol{Ax}=\boldsymbol{b}$ 的计算解. 简单地相减这两个等式并求解 $\delta\boldsymbol{x}$：一个做法是作

$$\begin{array}{rr} & (\boldsymbol{A}+\delta\boldsymbol{A})(\boldsymbol{x}+\delta\boldsymbol{x})=\boldsymbol{b}+\delta\boldsymbol{b} \\ - & [\boldsymbol{Ax}=\boldsymbol{b}] \\ \hline & \delta\boldsymbol{Ax}+(\boldsymbol{A}+\delta\boldsymbol{A})\delta\boldsymbol{x}=\delta\boldsymbol{b} \end{array}$$

整理得到

---

1. 例如，在第6章我们考虑逼近一个特殊的微分方程——泊松方程的解时产生的 $\boldsymbol{A}$ 的情况.

$$\delta x = A^{-1}(-\delta A\,\hat{x} + \delta b). \tag{2.1}$$

取范数并利用引理1.7的第1部分以及向量范数的三角形不等式，得到

$$\|\delta x\| \leqslant \|A^{-1}\|(\|\delta A\| \cdot \|\hat{x}\| + \|\delta b\|). \tag{2.2}$$

（在如1.7节中定义的，已假定向量范数和矩阵范数相容. 例如，向量范数及其导出的矩阵范数相容.）进一步整理该不等式得到

$$\frac{\|\delta x\|}{\|\hat{x}\|} \leqslant \|A^{-1}\| \cdot \|A\| \cdot \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \cdot \|\hat{x}\|}\right). \tag{2.3}$$

量 $\kappa(A) = \|A^{-1}\| \cdot \|A\|$ 是矩阵 A 的条件数，因为它度量的结果是解的相对改变 $\frac{\|\delta x\|}{\|\hat{x}\|}$ 为数据中的相对改变量 $\frac{\|\delta A\|}{\|A\|}$ 的倍数. [严格地讲，需要指出不等式(2.2)对某个非零的 δA 和 δb 等号成立；否则 $\kappa(A)$ 将仅仅是条件数的一个上界. 见问题2.3.] 若 δA 和 δb 是小的，则与 $\kappa(A)$ 相乘的量将是很小的，得到相对误差 $\frac{\|\delta x\|}{\|\hat{x}\|}$ 的一个小的上界.[1]

上界依赖于 δx（经由 $\hat{x}$），这使它似乎很难说明，但在实际上它是十分有用的，因为我们知道计算出的解 $\hat{x}$，故能直接地计算这个界. 我们还能导出一个理论上更吸引人的、不依赖于 δx 的界如下：

32

引理2.1 设 $\|\cdot\|$ 满足 $\|AB\| \leqslant \|A\| \cdot \|B\|$. 则 $\|X\| < 1$ 推出 $I - X$ 可逆，$(I-X)^{-1} = \sum_{i=0}^{\infty} X^i$，并且 $\|(I-X)^{-1}\| \leqslant \frac{1}{1-\|X\|}$.

证明 和 $\sum_{i=0}^{\infty} X^i$ 收敛当且仅当它按每个分量收敛. 利用事实（应用引理1.4到例1.6）对于每个范数，存在一个常数 c 使得 $|x_{jk}| \leqslant c \cdot \|X\|$. 得到 $|(X^i)_{jk}| \leqslant c \cdot \|X^i\| \leqslant c \cdot \|X\|^i$，故 $\sum X^i$ 的每个分量被一个收敛的几何级数 $\sum c\|X\|^i = \frac{c}{1-\|X\|}$ 所控制，则必定收敛. 所以当 $n \to \infty$ 时 $S_n = \sum_{i=0}^{n} X^i$ 收敛于某个 S，并且当 $n \to \infty$ 时，$(I-X)S_n = (I-X)(I+X+X^2+\cdots+X^n) = I - X^{n+1} \to I$，因为 $\|X^i\| \leqslant \|X\|^i \to 0$. 所以 $(I-X)S = I$，$S = (I-X)^{-1}$. 最后的界是 $\|(I-X)^{-1}\| = \left\|\sum_{i=0}^{\infty} X^i\right\| \leqslant \sum_{i=0}^{\infty}\|X^i\| \leqslant \sum_{i=0}^{\infty}\|X\|^i = \frac{1}{1-\|X\|}$. □

对第一个方程 $\delta Ax + (A+\delta A)\delta x = \delta b$ 求解 δx 得到

$$\begin{aligned}
\delta x &= (A+\delta A)^{-1}(-\delta Ax + \delta b) \\
&= [A(I + A^{-1}\delta A)]^{-1}(-\delta Ax + \delta b) \\
&= (I + A^{-1}\delta A)^{-1}A^{-1}(-\delta Ax + \delta b).
\end{aligned}$$

取范数，两边用 $\|x\|$ 相除，利用引理1.7的第1部分和三角形不等式，并假定 δA 足

1. 更严格地讲，它是关于矩阵求逆问题的条件数. 例如求 A 的特征值问题有不同的条件数.

够小使得$\|A^{-1}\delta A\| \le \|A^{-1}\| \cdot \|\delta A\| < 1$成立，即得所要求的界：

$$
\begin{aligned}
\frac{\|\delta x\|}{x} &\le \|(I+A^{-1}\delta A)^{-1}\| \cdot \|A^{-1}\| \left(\|\delta A\| + \frac{\|\delta b\|}{\|x\|} \right) \\
&\le \frac{\|A^{-1}\|}{1-\|A^{-1}\| \cdot \|\delta A\|} \left(\|\delta A\| + \frac{\|\delta b\|}{\|x\|} \right) \qquad \text{由引理 2.1} \\
&= \frac{\|A^{-1}\| \cdot \|A\|}{1-\|A^{-1}\| \cdot \|A\| \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \cdot \|x\|} \right) \\
&\le \frac{\kappa(A)}{1-\kappa(A) \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{b} \right) \qquad (2.4)
\end{aligned}
$$

因为$\|b\| = \|Ax\| \le \|A\| \cdot \|x\|$.

这个界把解中的相对误差表示为输入中的相对误差$\frac{\|\delta A\|}{\|A\|}$和$\frac{\|\delta b\|}{\|b\|}$的倍数. 当$\|\delta A\|$足够小时,乘子$\kappa(A) / \left(1-\kappa(A)\frac{\|\delta A\|}{\|A\|}\right)$接近于条件数$\kappa(A)$.

下面的定理将给出关于假定$\|A^{-1}\| \cdot \|\delta A\| = \kappa(A)\frac{\|\delta A\|}{\|A\|} < 1$的更多说明. 它保证$A+\delta A$非奇异，而这是$\delta x$存在所必需的. 它还建立了条件数的一个几何表示特征.

定理2.1　设A非奇异. 则

33

$$
\min\left\{ \frac{\|\delta A\|_2}{\|A\|_2} : A+\delta A \text{ 奇异} \right\} = \frac{1}{\|A^{-1}\|_2 \cdot \|A\|_2} = \frac{1}{\kappa(A)}.
$$

所以，到最接近的奇异矩阵的距离(不适定问题)$= \frac{1}{\text{条件数}}$.

证明　证明$\min\{\|\delta A\|_2 : A+\delta A \text{ 奇异}\} = \frac{1}{\|A^{-1}\|_2}$就足够了.

证明这个最小值至少是$\frac{1}{\|A^{-1}\|_2}$,注意，若$\|\delta A\|_2 < \frac{1}{\|A^{-1}\|_2}$,则$1 > \|\delta A\|_2 \cdot \|A^{-1}\|_2 \ge \|A^{-1}\delta A\|_2$,故由引理2.1推出$I+A^{-1}\delta A$可逆,故$A+\delta A$可逆.

证明这个最小值等于$\frac{1}{\|A^{-1}\|_2}$,构造一个范数为$\frac{1}{\|A^{-1}\|_2}$的δA使得$A+\delta A$奇异. 注意，因为$\|A^{-1}\|_2 = \max_{x\ne0}\frac{\|A^{-1}x\|_2}{\|x\|_2}$,所以存在一个$x$使得$\|x\|_2=1$且$\|A^{-1}\|_2 = \|A^{-1}x\|_2 > 0$. 现设$y = \frac{A^{-1}x}{\|A^{-1}x\|_2} = \frac{A^{-1}x}{\|A^{-1}\|_2}$,故$\|y\|_2=1$. 设$\delta A = \frac{-xy^{\mathrm{T}}}{\|A^{-1}\|_2}$,则

$$
\|\delta A\|_2 = \max_{z\ne0}\frac{\|xy^{\mathrm{T}}z\|_2}{\|A^{-1}\|_2\|z\|_2} = \max_{z\ne0}\frac{|y^{\mathrm{T}}z|\,\|x\|_2}{\|z\|_2\|A^{-1}\|_2} = \frac{1}{\|A^{-1}\|_2},
$$

当 z 是 y 的任意非零倍时，上式达到最大值，并且因为

$$(A+\delta A)y = Ay - \frac{xy^{T}y}{\|A^{-1}\|_2} = \frac{x}{\|A^{-1}\|_2} - \frac{x}{\|A^{-1}\|_2} = 0.$$

所以 $A+\delta A$ 是奇异的. □

现在我们已看到，到最接近的不适定问题的距离等于条件数的倒数，这一结论适用于两类问题：多项式求值和线性方程求解. 这种倒数关系在数值分析中是十分普遍的[71].

下面是研究 $Ax=b$ 扰动理论的一个方法，它稍稍不同于前面的方法. 我们需要用这个方法去导出后面 2.4.4 节中实际的误差界. 若 $\hat{x}$ 是一个任意的向量，可以如下求出差 $\delta x \equiv \hat{x}-x=\hat{x}-A^{-1}b$ 的界. 设 $r=A\hat{x}-b$ 是 $\hat{x}$ 的残差，当 $\hat{x}=x$ 时残差 r 为零. 记 $\delta x = A^{-1}r$，得到界

$$\|\delta x\| = \|A^{-1}r\| \leqslant \|A^{-1}\| \cdot \|r\|. \tag{2.5}$$

这个简单的界在实践中是有吸引力的，因为给定一个近似解 $\hat{x}$，r 是容易计算的. 此外，不存在明显的需要去估计 δA 和 δb. 事实上，正如下列定理指出的那样，我们的两个方法是非常密切相关的.

定理 2.2 设 $r=A\hat{x}-b$. 则存在一个 δA 使得 $\|\delta A\| = \frac{\|r\|}{\|\hat{x}\|}$ 且 $(A+\delta A)\hat{x}=b$. 不存在较小范数的 δA 且满足 $(A+\delta A)\hat{x}=b$. 因而，δA 是最小可能的（以范数度量的）向后误差. 这对任何向量范数及其导出范数（或者对向量 $\|\cdot\|_2$ 范数及对矩阵 $\|\cdot\|_F$ 范数）都是成立的.

34

证明 $(A+\delta A)\hat{x}=b$ 当且仅当 $\delta A\cdot\hat{x}=b-A\hat{x}=-r$，故 $\|r\|=\|\delta A\cdot\hat{x}\|\leqslant\|\delta A\|\cdot\|\hat{x}\|$，推得 $\|\delta A\|\geqslant\frac{\|r\|}{\|\hat{x}\|}$. 我们只对 2-范数及其导出范数完成证明. 选择 $\delta A=\frac{-r\cdot\hat{x}^{T}}{\|\hat{x}\|_2^2}$. 容易验证 $\delta A\cdot\hat{x}=-r$ 以及 $\|\delta A\|_2=\frac{\|r\|_2}{\|\hat{x}\|_2}$. □

因而，由定理 2.2 能得到一个满足 $(A+\delta A)\hat{x}=b$ 的最小的 $\|\delta A\|$ 并给出 $r=A\hat{x}-b$. 应用误差界(2.2)（对 $\delta b=0$）得到与(2.5)一样的界

$$\|\delta x\| \leqslant \|A^{-1}\|\left(\frac{\|r\|}{\|\hat{x}\|}\cdot\|\hat{x}\|\right) = \|A^{-1}\|\cdot\|r\|.$$

所有的界依赖于估计条件数 $\|A\|\cdot\|A^{-1}\|$ 的技巧. 在 2.4.3 节中将返回到这个问题. 条件数估计是用像 sgesvx 那样的 LAPACK 程序计算的.

相对扰动理论

在上一节中指出如何界定 $Ax=b$ 的近似解 $\hat{x}$ 中误差 $\delta x=\hat{x}-x$ 的范数. $\|\delta x\|$ 的界是与条件数 $\kappa(A)=\|A\|\cdot\|A^{-1}\|$ 与范数 $\|\delta A\|$ 和 $\|\delta b\|$ 的乘积成比例的，其中 $\hat{x}$ 满足 $(A+\delta A)\hat{x}=b+\delta b$.

在许多情形下这个界是令人满意的，但情况却不能总如此. 本节的目标是指出什么时候它不够好，并导出另一个提供比较严格的界的扰动理论. 在后面2.5.1节中将利用这个扰动理论去证明用像 sgesvx 那样的 LAPACK 子程序计算的误差界.

在首次阅读时本节可以跳过.

下面是上节中的误差界非常非常不好的一个例子.

例 2.1 设 $\boldsymbol{A}=\mathrm{diag}(\gamma,1)$（对角阵，对角元 $a_{11}=\gamma, a_{22}=1$），$\boldsymbol{b}=[\gamma,1]^{\mathrm{T}}$，其中 $\gamma>1$. 则 $\boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{b}=[1,1]^{\mathrm{T}}$. 任何适当的直接法将非常精确地求解 $\boldsymbol{Ax}=\boldsymbol{b}$（利用两次除法 b_i/a_{ii}）以得到 $\hat{\boldsymbol{x}}$，但是条件数 $\kappa(\boldsymbol{A})=\gamma$ 可能任意大. 所以误差界(2.3)可能任意大.

条件数 $\kappa(\boldsymbol{A})$ 导致我们过高估计误差的原因是它来自于界(2.2)，而后者假定 $\delta\boldsymbol{A}$ 是按范数有界的，但也是任意的. 问题2.3中需要这个条件证明界(2.2)可以达到. 相反，对应于实际舍入误差的 $\delta\boldsymbol{A}$ 不是任意的，但是它有不单独受其范数支配的一个特殊结构. 我们可以如下确定对应于我们问题中 $\hat{\boldsymbol{x}}$ 的最小 $\delta\boldsymbol{A}$：做个简单的舍入误差分析可得 $\hat{x}_i=(b_i/a_{ii})/(1+\delta_i)$，其中 $|\delta_i|\leqslant\varepsilon$. 因而 $(a_{ii}+\delta_i a_{ii})\hat{x}_i=b_i$. 可以重写此式为 $(\boldsymbol{A}+\delta\boldsymbol{A})\hat{\boldsymbol{x}}=\boldsymbol{b}$，其中 $\delta\boldsymbol{A}=\mathrm{diag}(\delta_1 a_{11},\delta_2 a_{22})$. 于是 $\|\delta\boldsymbol{A}\|$ 可以像 $\max_i|\varepsilon a_{ii}|=\varepsilon\gamma$ 那样大. 对 $\delta\boldsymbol{b}=0$ 应用误差界得到

[35]

$$\frac{\|\delta\boldsymbol{x}\|_\infty}{\|\hat{\boldsymbol{x}}\|_\infty}\leqslant\gamma\left(\frac{\varepsilon\gamma}{\gamma}\right)=\varepsilon\gamma.$$

相反，实际的误差满足

$$\begin{aligned}\|\delta\boldsymbol{x}\|_\infty&=\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|_\infty\\&=\left\|\begin{bmatrix}(b_1/a_{11})/(1+\delta_1)-(b_1/a_{11})\\(b_2/a_{22})/(1+\delta_2)-(b_2/a_{22})\end{bmatrix}\right\|_\infty\\&=\left\|\begin{bmatrix}-\delta_1/(1+\delta_1)\\-\delta_2/(1+\delta_2)\end{bmatrix}\right\|_\infty\leqslant\frac{\varepsilon}{1-\varepsilon}\end{aligned}$$

或

$$\frac{\|\delta\boldsymbol{x}\|_\infty}{\|\hat{\boldsymbol{x}}\|_\infty}\leqslant\varepsilon/(1-\varepsilon)^2,$$

结果小了 γ 倍. ◇

对此例可以描述实际的 $\delta\boldsymbol{A}$ 的结构如下：$|\delta a_{ij}|\leqslant\varepsilon|a_{ij}|$，其中 ε 是一个微小的数. 更简洁地记此式为

$$|\delta\boldsymbol{A}|\leqslant\varepsilon|\boldsymbol{A}| \tag{2.6}$$

（见1.1节的记号），也称 $\delta\boldsymbol{A}$ 是 $\boldsymbol{A}$ 中一个小的按分量相对扰动. 因为在实际上通常可以构造 $\delta\boldsymbol{A}$ 满足界(2.6)，以及 $|\delta\boldsymbol{b}|\leqslant\varepsilon|\boldsymbol{b}|$（见2.5.1节），我们将利用 $\delta\boldsymbol{A}$ 和 $\delta\boldsymbol{b}$ 的这些界推导扰动理论.

用(2.1)式开始：

$$\delta x = A^{-1}(-\delta A\hat{x} + \delta b).$$

对上式取绝对值并反复使用三角形不等式得到

$$\begin{aligned}|\delta x| &= |A^{-1}(-\delta A\hat{x} + \delta b)| \\ &\leqslant |A^{-1}|(|\delta A|\cdot|\hat{x}| + |\delta b|) \\ &\leqslant |A^{-1}|(\varepsilon|A|\cdot|\hat{x}| + \varepsilon|b|) \\ &= \varepsilon(|A^{-1}|(|A|\cdot|\hat{x}| + |b|)).\end{aligned}$$

利用任何满足$\||z|\| = \|z\|$的向量范数（像无穷范数，1－范数，弗罗贝尼乌斯范数），得到界

$$\|\delta x\| \leqslant \varepsilon\||A^{-1}|(|A|\cdot|\hat{x}| + |b|)\|. \tag{2.7}$$

36

暂时假定$\delta b = 0$，可以把这个界弱化为

$$\|\delta x\| \leqslant \varepsilon\||A^{-1}|\cdot|A|\|\cdot\|\hat{x}\|$$

或

$$\frac{\|\delta x\|}{\|x\|} \leqslant \varepsilon\||A^{-1}|\cdot|A|\|. \tag{2.8}$$

这引导我们定义$\kappa_{CR}(A) \equiv \||A^{-1}|\cdot|A|\|$为$A$的按分量相对条件数，或就简称为相对条件数，有时也称之为Bauer条件数[26]或Skeel条件数[225,226,227]. 证明(2.7)和(2.8)的界是可以达到的，见问题2.4.

回顾条件数$\kappa(A)$与从A到最接近的奇异阵的距离有关的定理2.1. 关于$\kappa_{CR}(A)$的一个类似的说明见[72,208].

例2.2 考虑以前的例子，$A = \mathrm{diag}(\gamma, 1)$和$b = [\gamma, 1]^T$. 容易确定$\kappa_{CR}(A) = 1$，因为$|A^{-1}||A| = I$. 实际上，对任意的对角阵$A$，$\kappa_{CR}(A) = 1$，直觉上提示我们，一个对角方程组应该能非常精确地求解. ◇

更一般地，假定D是任意非奇异对角阵而B是一个任意的非奇异阵，则

$$\begin{aligned}\kappa_{CR}(DB) &= \||(DB)^{-1}|\cdot|(DB)|\| \\ &= \||B^{-1}D^{-1}|\cdot|DB|\| \\ &= \||B^{-1}|\cdot|B|\| \\ &= \kappa_{CR}(B).\end{aligned}$$

这意味着若DB是坏的缩放，即B是良态的而DB是坏的(因为D有相差很大的对角元)，则尽管DB是坏的，我们仍期望得到$(DB)x = b$的一个精确解. 这在2.4.4节、2.5.1节和2.5.2节中进一步讨论.

最后，如上节中那样，只利用残差$r = A\hat{x} - b$提供一个误差界：

$$\|\delta x\| = \|A^{-1}r\| \leqslant \||A^{-1}|\cdot|r|\|, \tag{2.9}$$

这里我们利用了三角形不等式. 在2.4.4节中将看到，特别当A是坏的缩放时，这个界有时能比类似的界(2.5)小得多. 此外，还存在一个类似于定理2.2的结论[193].

定理2.3 使$|\delta A| \leqslant \varepsilon|A|$和$|\delta b| \leqslant \varepsilon|b|$成立且满足$(A + \delta A)\hat{x} = b + \delta b$的最小的$\varepsilon > 0$,

称为按分量的相对向后误差. 它可以根据残差 $r = A\hat{x} - b$ 表达如下：

37

$$\varepsilon = \max_i \frac{|r_i|}{(|A|\cdot|\hat{x}|+|b|)_i}.$$

证明见问题2.5.

像 sgesvx 那样的 LAPACK 程序可以计算按分量的向后相对误差 ε（ε 的 LAPACK 变量名是 BERR）.

2.3　高斯消元法

求解 $Ax = b$ 的基本算法是高斯消元法. 为叙述它，需要先定义置换矩阵.

定义2.1　置换矩阵 P 是一个带有置换行的单位矩阵.

下列引理给出置换阵最重要的性质.

引理2.2　设 P、P_1 和 P_2 是 $n\times n$ 阶置换阵，X 是一个 $n\times n$ 阶矩阵，则

1. PX 等同于带有置换行的 X；XP 等同于带有置换列的 X.
2. $P^{-1} = P^{\mathrm{T}}$.
3. $\det(P) = \pm 1$.
4. $P_1 \cdot P_2$ 也是一个置换阵.

证明见问题2.6.

现在可叙述求解 $Ax = b$ 完整的算法.

> **算法2.1**　利用高斯消元法解 $Ax = b$：
>
> 1. 将 A 分解为 $A = PLU$，其中
>   - P = 置换矩阵
>   - L = 单位下三角形阵（即在对角线上为1）
>   - U = 非奇异上三角形阵
> 2. 通过置换 b 的元素，求解 $PLUx = b$ 得 LUx：$LUx = P^{-1}b = P^{\mathrm{T}}b$.
> 3. 通过向前回代求解 $LUx = P^{-1}b$ 得 Ux：$Ux = L^{-1}(P^{-1}b)$.
> 4. 通过向后回代求解 $Ux = L^{-1}(P^{-1}b)$ 得 x：$x = U^{-1}(L^{-1}P^{-1}b)$.

我们将用几种方式导出分解 $A = PLU$ 的算法. 我们首先说明为什么置换矩阵 P 是必要的.

38

定义2.2　A 的前 $j\times j$ 阶主子阵是 $A(1:j,1:j)$.

定理2.4　下列两个语句是等价的：

1. 存在唯一的下三角形阵 L 和非奇异上三角形阵 U 使得 $A = LU$.
2. A 的所有前主子阵非奇异.

证明　首先证明（1）推出（2）. $A = LU$ 也可写成

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} \end{bmatrix},$$

其中 A_{11} 是同 L_{11} 和 U_{11} 一样的 $j \times j$ 阶前主子阵. 因为 L 是单位下三角形阵, U 是非奇异上三角形阵, 所以 $\det A_{11} = \det(L_{11}U_{11}) = \det L_{11} \det U_{11} = 1 \cdot \prod_{k=1}^{j}(U_{11})_{kk} \neq 0$.

对 n 用归纳法, 证明 (2) 推出 (1). 对 1×1 阶矩阵容易证明: $a = 1 \cdot a$. 为了对 $n \times n$ 阵 $\widetilde{A}$ 证明它, 需要求唯一的 $(n-1) \times (n-1)$ 三角形阵 L 和 U, 唯一的 $(n-1) \times 1$ 向量 l 和 u, 以及唯一的非零标量 η 使得

$$\widetilde{A} = \begin{bmatrix} A & b \\ c^{\mathrm{T}} & \delta \end{bmatrix} = \begin{bmatrix} L & 0 \\ l^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} U & u \\ 0 & \eta \end{bmatrix} = \begin{bmatrix} LU & Lu \\ l^{\mathrm{T}}U & l^{\mathrm{T}}u + \eta \end{bmatrix}.$$

由归纳法知, 存在唯一的 L 和 U 使得 $A = LU$. 今设 $u = L^{-1}b, l^{\mathrm{T}} = c^{\mathrm{T}}U^{-1}$ 以及 $\eta = \delta - l^{\mathrm{T}}u$, 这些都是唯一的. 由归纳法知 U 的对角元都非零, 并且因为 $0 \neq \det(\widetilde{A}) = \det(U) \cdot \eta$, 所以 $\eta \neq 0$. □

因而对于像置换阵

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

那样(良态的)非奇异阵, 不选主元的 LU 分解可能失败, 因为 P 的 1×1 阶和 2×2 阶主子阵是奇异的. 故我们需要在高斯消元法中引进置换.

定理 2.5 *若 A 非奇异, 则存在置换 P_1 和 P_2, 一个单位下三角形阵 L 和一个非奇异上三角形阵 U, 使得 $P_1AP_2 = LU$. P_1 和 P_2 中只有一个是必要的.*

注: P_1A 将 A 的行重新排序, AP_2 将 A 的列重新排序, 而 P_1AP_2 同时重新排序 A 的行和列.

39

证明 与许多矩阵分解一样, 弄懂 2×2 阶矩阵块就足够了, 然后对维数 n 用归纳法. 对 1×1 阶矩阵是容易的: $P_1 = P_2 = L = 1, U = A$. 假定结论对 $n-1$ 维成立. 若 A 非奇异, 则它有一个非零元; 选择置换 P'_1 和 P'_2 使得 $P'_1AP'_2$ 的 $(1,1)$ 元非零. (只需要 P'_1 和 P'_2 之一, 因为由非奇异性推出 A 的每一行和每一列有一个非零元.)

现在写出所要求的分解并求解未知的分量:

$$P'_1AP'_2 = \begin{bmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ L_{21} & I \end{bmatrix} \cdot \begin{bmatrix} u_{11} & U_{12} \\ 0 & \widetilde{A}_{22} \end{bmatrix} = \begin{bmatrix} u_{11} & U_{12} \\ L_{21}u_{11} & L_{21}U_{12} + \widetilde{A}_{22} \end{bmatrix}, \quad (2.10)$$

其中 A_{22} 和 $\widetilde{A}_{22}$ 是 $(n-1) \times (n-1)$ 阶矩阵, L_{21} 和 U_{12}^{T} 是 $(n-1) \times 1$ 阶矩阵.

求解这个 2×2 块分解的块分量, 我们得到 $u_{11} = a_{11} \neq 0, U_{12} = A_{12}, L_{21}u_{11} = A_{21}$. 因为 $u_{11} = a_{11} \neq 0$, 可解得 $L_{21} = \frac{A_{21}}{a_{11}}$. 最后, $L_{21}U_{12} + \widetilde{A}_{22} = A_{22}$ 推出 $\widetilde{A}_{22} = A_{22} - L_{21}U_{12}$.

要对 $\widetilde{A}_{22}$用归纳法，为此需要检查 det $\widetilde{A}_{22}\neq 0$：因为 $\det P'_1AP'_2=\pm\det A\neq 0$ 并且

$$\det P'_1AP'_2=\det\begin{bmatrix}1 & 0\\ L_{21} & I\end{bmatrix}\cdot\det\begin{bmatrix}u_{11} & U_{12}\\ 0 & \widetilde{A}_{22}\end{bmatrix}=1\cdot(u_{11}\cdot\det\widetilde{A}_{22}),$$

于是 det $\widetilde{A}_{22}$必为非零.

所以，由归纳法知存在置换 $\widetilde{P}_1$和 $\widetilde{P}_2$使得 $\widetilde{P}_1\widetilde{A}_{22}\widetilde{P}_2=\widetilde{L}\widetilde{U}$，$\widetilde{L}$ 为单位下三角形阵和 $\widetilde{U}$ 为非奇异上三角形阵. 把这个分解代入到上面 2×2 块分解中得到

$$\begin{aligned}P'_1AP'_2&=\begin{bmatrix}1 & 0\\ L_{21} & I\end{bmatrix}\begin{bmatrix}u_{11} & U_{12}\\ 0 & \widetilde{P}_1^{\mathrm{T}}\widetilde{L}\widetilde{U}\widetilde{P}_2^{\mathrm{T}}\end{bmatrix}\\
&=\begin{bmatrix}1 & 0\\ L_{21} & I\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \widetilde{P}_1^{\mathrm{T}}\widetilde{L}\end{bmatrix}\begin{bmatrix}u_{11} & U_{12}\\ 0 & \widetilde{U}\widetilde{P}_2^{\mathrm{T}}\end{bmatrix}\\
&=\begin{bmatrix}1 & 0\\ L_{21} & \widetilde{P}_1^{\mathrm{T}}\widetilde{L}\end{bmatrix}\begin{bmatrix}u_{11} & U_{12}\widetilde{P}_2\\ 0 & \widetilde{U}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \widetilde{P}_2^{\mathrm{T}}\end{bmatrix}\\
&=\begin{bmatrix}1 & 0\\ 0 & \widetilde{P}_1^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}1 & 0\\ \widetilde{P}_1L_{21} & \widetilde{L}\end{bmatrix}\begin{bmatrix}u_{11} & U_{12}\widetilde{P}_2\\ 0 & \widetilde{U}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \widetilde{P}_2^{\mathrm{T}}\end{bmatrix},\end{aligned}$$

因而我们得到所要求的 A 的分解：

$$\begin{aligned}P_1AP_2&=\left(\begin{bmatrix}1 & 0\\ 0 & \widetilde{P}_1\end{bmatrix}P'_1\right)A\left(P'_2\begin{bmatrix}1 & 0\\ 0 & \widetilde{P}_2\end{bmatrix}\right)\\
&=\begin{bmatrix}1 & 0\\ \widetilde{P}_1L_{21} & \widetilde{L}\end{bmatrix}\begin{bmatrix}u_{11} & U_{12}\widetilde{P}_2\\ 0 & \widetilde{U}\end{bmatrix}.\end{aligned}$$

40 □

下面两个推论叙述的是选择 P_1 和 P_2 的简单方法，以保证高斯消元法对一个非奇异阵成功.

推论2.1 选择 $P'_2=I$ 和 P'_1 使得 a_{11}是它所在列中绝对值最大的元素，推出 $L_{21}=\frac{A_{21}}{a_{11}}$ 的元素绝对值以1为界. 更一般地，在高斯消元法的第 i 步，在计算 L 的第 i 列时，将第 i 行到第 n 行重新排序使得该列的最大元在对角线上，这称为“部分选主元的高斯消元法”或简称为 GEPP. GEPP 保证 L 的所有元素之绝对值以1为界.

实际上 GEPP 是实现高斯消元法最普通的方式. 在下节中将讨论它的数值稳定

性. 另一个选择 $\boldsymbol{P}_1$ 和 $\boldsymbol{P}_2$ 的更费时的方式由下面的推论给出. 该方式在实践中很少用到，虽然有少数例子反映出在计算精确值时，GEPP 方式失败，而这一方式却成功了（见问题2.14）. 我们在下一节将会简要讨论此问题.

推论2.2 选择 $\boldsymbol{P}'_1$ 和 $\boldsymbol{P}'_2$ 使得 a_{11} 是整个矩阵中绝对值最大的元素. 更一般地，在高斯消元法的第 i 步，在计算 $\boldsymbol{L}$ 的第 i 列时，将第 i 行到第 n 行和第 i 列到第 n 列重新排序使得这个子矩阵中的最大元在对角线上. 这称为"完全选主元的高斯消元法"或简称为 GECP.

下列算法具体表达定理2.5，执行置换，计算 $\boldsymbol{L}$ 的第一列和 $\boldsymbol{U}$ 的第一行，更新 $\boldsymbol{A}_{22}$ 得到 $\widetilde{\boldsymbol{A}}_{22}=\boldsymbol{A}_{22}-\boldsymbol{L}_{21}\boldsymbol{U}_{12}$. 先用常规的程序设计语言符号，再用 Matlab 符号编写算法.

算法2.2 选主元的 $\boldsymbol{LU}$ 分解：

```
for i = 1 to n - 1
    应用置换以使 a_ii ≠ 0(也置换 L 和 U)
            /* 例如，对 GEPP，交换 A 的第 j 行和第 i 行，以及 L 的第
              j 行和第 i 行，其中 |a_ji| 是 |A(i:n,i)| 中的最大元；
              对 GECP，交换 A 的第 j 行和第 i 行，以及 L 的第 j 行和
              第 i 行，交换 A 的第 k 列和第 i 列，以及 U 的第 k 列和第
              i 列，其中 |a_jk| 是 |A(i:n,i:n)| 中的最大元 */
    /* 计算 L 的第 i 列((2.10)中的 L_21) */
    for j = i + 1 to n
        l_ji = a_ji / a_ii
    end for
    /* 计算 U 的第 i 行((2.10)中的 U_12) */
    for j = i to n
        u_ij = a_ij
    end for
    /* 更新 A_22(得到(2.10)中的 Ã_22 = A_22 - L_21 U_12) */
    for j = i + 1 to n
        for k = i + 1 to n
            a_jk = a_jk - l_ji * u_ik
        end for
    end for
end for
```

41

注意一旦 $\boldsymbol{A}$ 的第 i 列被用于计算 $\boldsymbol{L}$ 的第 i 列之后，它将绝不再使用. 类似的 $\boldsymbol{A}$ 的第 i 行在计算 $\boldsymbol{U}$ 的第 i 行之后将绝不再使用. 这就允许我们当 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 算出后把 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 覆盖在 $\boldsymbol{A}$ 的上部，因而为了存储它们不需要额外的空间. $\boldsymbol{L}$ 占用 $\boldsymbol{A}$ 的(严格)下三角形($\boldsymbol{L}$ 对角线上的1不是明显地存放的)，而 $\boldsymbol{U}$ 占用 $\boldsymbol{A}$ 的上三角形，这就将其简化为下列算法.

算法 2.3　选主元的 $\boldsymbol{LU}$ 分解，$\boldsymbol{L}$ 和 $\boldsymbol{U}$ 覆盖在 $\boldsymbol{A}$ 上：

```
for i = 1 to n - 1
    应用置换(细节见算法 2.2)
    for j = i + 1 to n
        a_ji = a_ji / a_ii
    end for
    for j = i + 1 to n
        for k = i + 1 to n
            a_jk = a_jk - a_ji * a_ik
        end for
    end for
end for
```

利用 Matlab 符号，这个算法进一步化简为下列算法.

算法 2.4　选主元的 $\boldsymbol{LU}$ 分解，$\boldsymbol{L}$ 和 $\boldsymbol{U}$ 覆盖在 $\boldsymbol{A}$ 上：

```
for i = 1 to n - 1
    应用置换(细节见算法 2.2)
    A(i+1 : n,i) = A(i+1 : n,i)/A(i,i)
    A(i+1 : n,i+1 : n) =
        A(i+1 : n,i+1 : n) - A(i+1 : n,i) * A(i,i+1 : n)
end for
```

在算法的最后行中，$\boldsymbol{A}(i+1:n,i) * \boldsymbol{A}(i,i+1:n)$ 是 $(n-i)\times 1$ 阶矩阵($\boldsymbol{L}_{21}$)与 [42] $1\times(n-i)$ 阶矩阵($\boldsymbol{U}_{12}$)之积，它得到一个 $(n-i)\times(n-i)$ 阶矩阵.

我们现在从头做起，从或许最为熟悉的高斯消元法的描述重新导出这个算法. "取每行并从后面的行中减去该行的倍数，使对角线下的元素全化为零." 把这个方法直接转变成一个算法得到

```
for i = 1 to n - 1                /* 对每 i 行 */
    for j = i + 1 to n            /* 从第 j 行中减去第 i 行的倍数…… */
        for k = i to n            /* ……在第 i 列直到第 n 列中…… */
```

```
      a_jk = a_jk - (a_ji / a_ii) a_ik       /* 将第 i 列对角线下面元素全化为零 */
    end for
  end for
end for
```

现在对这个算法作某些改进，对它修改直到它变成和算法 2.3 一致(选主元除外). 首先，认识到不需要计算对角线下面的零元，因为我们知道它们为零，这就缩短了 k 循环得到

```
for i = 1 to n - 1
  for j = i + 1 to n
    for k = i + 1 to n
      a_jk = a_jk - (a_ji / a_ii) a_ik
    end for
  end for
end for
```

下一个进行的改进是在内循环之外计算$\frac{a_{ji}}{a_{ii}}$，因为在内循环之中它是常数.

```
for i = 1 to n - 1
  for j = i + 1 to n
    l_ji = a_ji / a_ii
  end for
  for j = i + 1 to n
    for k = i + 1 to n
      a_jk = a_jk - l_ji a_ik
    end for
  end for
end for
```

最后，把乘子 l_{ji}存放在原来化为零的次对角元 a_{ji}位置上. 因为不再用到它们. 这就得到算法 2.3(选主元除外).

43

在相同的范围内用加法代替循环，而内循环用它们的运算量求出 **LU** 分解的运算量:

$$\sum_{i=1}^{n-1}\left(\sum_{j=i+1}^{n}1+\sum_{j=i+1}^{n}\sum_{k=i+1}^{n}2\right)$$
$$=\sum_{i=1}^{n-1}\left((n-i)+2(n-i)^{2}\right)=\frac{2}{3}n^{3}+O(n^{2}).$$

用$\boldsymbol{L}$和$\boldsymbol{U}$的向前回代和向后回代去求解$\boldsymbol{Ax}=\boldsymbol{b}$的工作量为$O(n^2)$，故利用高斯消元法求解$\boldsymbol{Ax}=\boldsymbol{b}$的全部的工作量为$\frac{2}{3}n^3+O(n^2)$次运算. 这里利用了结论$\sum_{i=1}^{m}i^k=m^{k+1}/(k+1)+O(m^k)$. 这样足以得到运算量的高阶项.

除了算法2.2书写的嵌套循环之外，实施高斯消元法还要做更多的工作. 实际上，随着计算机、程序设计语言以及矩阵大小的不同，仅仅交换最后的关于j和k的两个循环，执行时间就会发生数量级的改变. 在2.6节再来讨论这个问题.

2.4　误差分析

回顾得到$\boldsymbol{Ax}=\boldsymbol{b}$解的误差界的两步范例：

1. 分析舍入误差得到$\boldsymbol{Ax}=\boldsymbol{b}$的解是扰动的线性方程组$(\boldsymbol{A}+\delta\boldsymbol{A})\hat{\boldsymbol{x}}=\boldsymbol{b}+\delta\boldsymbol{b}$的精确解，其中$\delta\boldsymbol{A}$和$\delta\boldsymbol{b}$是小的. 这是*向后误差分析*的一个例子，$\delta\boldsymbol{A}$和$\delta\boldsymbol{b}$称为*向后误差*.

2. 应用2.2节的扰动理论去界定误差，例如利用界(2.3)或(2.5).

本节有两个目标. 第一个是指出为了保持向后误差$\delta\boldsymbol{A}$和$\delta\boldsymbol{b}$是小的，如何实施高斯消元法. 特别地，希望保持$\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}$和$\frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}$像$O(\varepsilon)$那样小. 我们可以使其如期望的那样小，因为仅仅舍入$\boldsymbol{A}$(或$\boldsymbol{b}$)的最大元去适合浮点格式就可以使得$\frac{\|\delta\boldsymbol{A}\|}{\boldsymbol{A}}\geqslant\varepsilon$(或$\frac{\|\delta\boldsymbol{b}\|}{\boldsymbol{b}}\geqslant\varepsilon$). 我们发觉，除非小心地选主元，否则$\delta\boldsymbol{A}$和$\delta\boldsymbol{b}$不一定是小的. 我们在下节会讨论这个问题.

第二个目标是导出计算不费力且“紧凑的”(即接近于真正的误差)的实用误差界. 结果是我们能正式证明的$\|\delta\boldsymbol{A}\|$的最好的界一般比实际出现的误差大得多. 所以实用的误差界(2.4.4节中)将依赖于计算的残差$\boldsymbol{r}=\boldsymbol{A}\hat{\boldsymbol{x}}-\boldsymbol{b}$和界(2.5)，而不是界(2.3). 我们也需要能够不费力地估计$\kappa(A)$，这在2.4.3节中讨论.

44

遗憾的是，始终没有满足廉价性和紧凑性两个目标的误差界，即它同时满足：

1. 首先与解$\boldsymbol{Ax}=\boldsymbol{b}$相比花费的工作量是可以忽略不计的(例如，花费$O(n^2)$ flops，而高斯消元法花费$O(n^3)$ flops).

2. 提供的误差界至少总是像真正的误差那样大，并且决不会超过它的常数倍(譬如比100倍更大).

2.4.4节中的实用的误差界将花费$O(n^2)$，但这只有在非常罕见的场合提供过分小或过分大的误差界时才出现. 得到一个坏的误差界的概率非常之小，所以在实践中这些界被广泛地使用. 唯一真正地有保证的界或者使用区间运算或者使用非常高精度的运算，或者两者同时使用，它比单纯求解$\boldsymbol{Ax}=\boldsymbol{b}$费时多好几倍(见1.5节).

人们曾猜测，事实上不存在同时满足廉价性和紧凑性目标的界，这至今仍为一个悬而未决的问题.

2.4.1 选主元的必要性

对 $A=\begin{bmatrix}0.0001 & 1\\ 1 & 1\end{bmatrix}$按三位十进制浮点算术运算应用不选主元的 **LU** 分解，看看为什么会得出错误的答案. 注意 $\kappa(A)=\|A\|_\infty\cdot\|A^{-1}\|_\infty\approx 4$,故 A 是良态的，因而能期望精确地求解 $Ax=b$.

$$L=\begin{bmatrix}1 & 0\\ \mathrm{fl}(1/10^{-4}) & 1\end{bmatrix},\mathrm{fl}(1/10^{-4})\text{ 舍入到 }10^4,$$

$$U=\begin{bmatrix}10^{-4} & 1\\ & \mathrm{fl}(1-10^4\cdot 1)\end{bmatrix},\mathrm{fl}(1-10^4\cdot 1)\text{ 舍入到 }-10^4,$$

故 $$LU=\begin{bmatrix}1 & 0\\ 10^4 & 1\end{bmatrix}\begin{bmatrix}10^{-4} & 1\\ & -10^4\end{bmatrix}=\begin{bmatrix}10^{-4} & 1\\ 1 & 0\end{bmatrix}$$

但是 $$A=\begin{bmatrix}10^{-4} & 1\\ 1 & 1\end{bmatrix}.$$

注意原来的 a_{22}从它减去 10^4 的计算中完全地“丢失”. 不管 a_{22}为 1，0，-2 或者任何使 $\mathrm{fl}(a_{22}-10^4)=-10^4$ 的数，我们将得出同样的 **LU** 因子. 因为算法仅对 L 和 U 进行操作，所以对应于不同的 a_{22}（从而对应于完全不同的 A 和完全不同的 $x=A^{-1}b$）将得到相同的答案，但不保证得到一个精确的答案. 这称为数值不稳定性，因为 L 和 U 不是接近于 A 的矩阵的因子.（解释这种情况的另一种方式是 $\|A-LU\|$大约像 $\|A\|$那样大而不是 $\varepsilon\|A\|$那样大.）

45

我们观察当利用这个 **LU** 分解求解 $Ax=[1,2]^T$ 的 x 时发生什么情况. 正确的答案应该是 $x\approx[1,1]^T$. 但我们获得下列结果. 解 $Ly=[1,2]^T$ 得到 $y_1=\mathrm{fl}(1/1)=1$ 和 $y_2=\mathrm{fl}(2-10^4\cdot 1)=-10^4$,注意值 2 从它减去 10^4 时被“丢失”了. 解 $U\hat{x}=y$ 得到 $\hat{x}_2=\mathrm{fl}((-10^4)/(-10^4))=1$ 和 $\hat{x}_1=\mathrm{fl}((1-1)/10^{-4})=0$,这是一个完全不正确的解.

精确度丧失的另一种预兆来自于 A 的条件数与 L 和 U 的条件数的比较. 此前我们把求解 $Ax=b$ 的问题转换为求解其他两个关于 L 和 U 的方程组，故我们不希望 L 和 U 的条件数比 A 的条件数大很多. 但是这里 A 的条件数大约是 4，而 L 和 U 的条件数大约是 10^8.

在下节中将说明执行 GEPP 总可以消除刚才说明的不稳定性. 在上例中，GEPP 在方法进行之前对换了两个方程的次序. 请读者确认此时我们将得到

$$L=\begin{bmatrix}1 & 0\\ \mathrm{fl}(0.0001/1) & 1\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0.0001 & 1\end{bmatrix}$$

和

$$U=\begin{bmatrix}1 & 1\\ 0 & \mathrm{fl}(1-0.0001\cdot 1)\end{bmatrix}=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}.$$

所以 LU 十分精确地近似于 A. L 和 U 两者像 A 一样都是十分良态的. 计算的解向量也是十分精确的.

2.4.2 高斯消元法正式的误差分析

下面是 LU 分解的误差分析之后的直观感觉. 若乘积 LU 中产生的中间量与 $\|A\|$相比较非常大, 当从它们中减去这些大的值时 A 的元素中的信息将“丢失”, 这就是2.4.1节的例中 a_{22}所遇到的问题. 假若乘积 LU 中的中间量与 A 相当, 则可以期望在分解中产生一个微小的向后误差 $A-LU$. 所以, 我们要界定乘积 LU 中的最大的中间量. 我们通过界定矩阵$|L|\cdot|U|$的元来做这件事(记号见1.1节).

我们的分析类似于1.6节中对多项式求值所使用的方法. 那里考察 $p=\sum_i a_i x^i$ 并指出若$|p|$与绝对值之和 $\sum_i |a_i x^i|$ 相当, 则 p 将被精确地算出.

在提出高斯消元法的一个一般的分析之后, 将利用它去证明 GEPP(或更费时的 GECP)在几乎所有的实际情况中将保持$|L|\cdot|U|$的元素与$\|A\|$相当.

[46]

遗憾的是, 可以证明的一般$\|\delta A\|$的最优的界仍旧比实际中出现的误差大得多. 所以, 我们在实际中使用的误差界是基于计算残差 r 和界(2.5)[或界(2.9)]的, 而不是本节中的严格的然而悲观的界.

现假定矩阵 A 已选定主元, 这样记号更为简单. 简化算法2.2为两个等式, 一个是关于 $a_{jk},j\le k$,另一个是关于 $a_{jk},j>k$. 首先追溯当 $j\le k$ 时算法2.2对 a_{jk}的运算: 对 $i=1$ 到 $j-1$, 这个元素被反复减去 $l_{ji}u_{ik}$并且最后赋值给 u_{jk}使得

$$u_{jk}=a_{jk}-\sum_{i=1}^{j-1} l_{ji}u_{ik}.$$

当 $j>k$ 时, 对 $i=1$ 到 $k-1$, a_{jk}再被反复减去 $l_{ji}u_{ik}$, 所得的差除以 u_{kk}并且赋值给 l_{jk}:

$$l_{jk}=\frac{a_{jk}-\sum_{i=1}^{k-1} l_{ji}u_{ik}}{u_{kk}}.$$

对这两个公式做舍入误差分析, 利用问题1.10的结论, 按浮点算术运算计算的点积满足

$$\mathrm{fl}\left(\sum_{i=1}^{d} x_i y_i\right)=\sum_{i=1}^{d} x_i y_i(1+\delta_i),\ |\delta_i|\le d\varepsilon.$$

对 u_{jk}的公式应用上式得到[1]

$$u_{jk}=\left(a_{jk}-\sum_{i=1}^{j-1} l_{ji}u_{ik}(1+\delta_i)\right)(1+\delta'),$$

---

1. 严格地讲, 下列公式假定先求和再从 a_{jk}中减去, 而最后的界不依赖于求和的次序.

其中$|\delta_i|\leqslant(j-1)\varepsilon$，$|\delta'|\leqslant\varepsilon$. 解 a_{jk}得到

$$a_{jk}=\frac{1}{1+\delta'}u_{jk}\cdot l_{jj}+\sum_{i=1}^{j-1}l_{ji}u_{ik}(1+\delta_i)\qquad \text{因为 } l_{jj}=1$$

$$=\sum_{i=1}^{j}l_{ji}u_{ik}+\sum_{i=1}^{j}l_{ji}u_{ik}\delta_i$$

其中$|\delta_i|\leqslant(j-1)\varepsilon$,且$1+\delta_j\equiv\frac{1}{1+\delta'}$

$$\equiv\sum_{i=1}^{j}l_{ji}u_{ik}+E_{jk},$$

47

其中 $\boldsymbol{E}_{jk}$的界为

$$|\boldsymbol{E}_{jk}|=\left|\sum_{i=1}^{j}l_{ji}\cdot u_{ik}\cdot\delta_i\right|\leqslant\sum_{i=1}^{j}|l_{ji}|\cdot|u_{ik}|\cdot n\,\varepsilon=n\,\varepsilon(|\boldsymbol{L}|\cdot|\boldsymbol{U}|)_{jk}.$$

对求 l_{jk}的公式做同样的分析得到

$$l_{jk}=(1+\delta'')\left(\frac{(1+\delta')(a_{jk}-\sum_{i=1}^{k-1}l_{ji}u_{ik}(1+\delta_i))}{u_{kk}}\right),$$

其中$|\delta_i|\leqslant(k-1)\varepsilon$,$|\delta'|\leqslant\varepsilon$,$|\delta''|\leqslant\varepsilon$. 解 a_{jk}得到

$$a_{jk}=\frac{1}{(1+\delta')(1+\delta'')}u_{kk}l_{jk}+\sum_{i=1}^{k-1}l_{ji}u_{ik}(1+\delta_i)$$

$$=\sum_{i=1}^{k}l_{ji}u_{ik}+\sum_{i=1}^{k}l_{ji}u_{ik}\delta_i\qquad \text{其中 } 1+\delta_k\equiv\frac{1}{(1+\delta')(1+\delta'')}$$

$$\equiv\sum_{i=1}^{k}l_{ji}u_{ik}+E_{jk},$$

其中$|\delta_i|\leqslant n\,\varepsilon$,故由前可知$|\boldsymbol{E}_{jk}|\leqslant n\,\varepsilon(|\boldsymbol{L}|\cdot|\boldsymbol{U}|)_{jk}$.

总而言之，我们可以用简单公式$\boldsymbol{A}=\boldsymbol{LU}+\boldsymbol{E}$，其中$|\boldsymbol{E}|\leqslant n\,\varepsilon|\boldsymbol{L}|\cdot|\boldsymbol{U}|$来概括误差分析. 取范数得到$\|\boldsymbol{E}\|\leqslant n\varepsilon\|\,|\boldsymbol{L}|\,\|\cdot\|\,|\boldsymbol{U}|\,\|$. 若范数不依赖于矩阵元素的符号(对弗罗贝尼乌斯范数，无穷范数和1-范数成立，但对2-范数不成立)，则可以简化成$\|\boldsymbol{E}\|\leqslant n\varepsilon\|\boldsymbol{L}\|\cdot\|\boldsymbol{U}\|$.

现考虑问题的其余部分：由 $\boldsymbol{Ly}=\boldsymbol{b}$ 和 $\boldsymbol{Ux}=\boldsymbol{y}$ 求解 $\boldsymbol{LUx}=\boldsymbol{b}$. 问题 1.11 的结论指出，用向前回代解 $\boldsymbol{Ly}=\boldsymbol{b}$ 得到一个满足$(\boldsymbol{L}+\delta\boldsymbol{L})\hat{\boldsymbol{y}}=\boldsymbol{b}$ 的计算解 $\hat{\boldsymbol{y}}$，其中$|\delta\boldsymbol{L}|\leqslant n\varepsilon|\boldsymbol{L}|$. 类似地，解$\boldsymbol{Ux}=\hat{\boldsymbol{y}}$时，得到满足$(\boldsymbol{U}+\delta\boldsymbol{U})\hat{\boldsymbol{x}}=\hat{\boldsymbol{y}}$的 $\hat{\boldsymbol{x}}$，其中$|\delta\boldsymbol{U}|\leqslant n\varepsilon|\boldsymbol{U}|$.

合并这些结果得到

$$\begin{aligned}\boldsymbol{b}&=(\boldsymbol{L}+\delta\boldsymbol{L})\hat{\boldsymbol{y}}\\&=(\boldsymbol{L}+\delta\boldsymbol{L})(\boldsymbol{U}+\delta\boldsymbol{U})\hat{\boldsymbol{x}}\\&=(\boldsymbol{LU}+\boldsymbol{L}\delta\boldsymbol{U}+\delta\boldsymbol{LU}+\delta\boldsymbol{L}\delta\boldsymbol{U})\hat{\boldsymbol{x}}\\&=(\boldsymbol{A}-\boldsymbol{E}+\boldsymbol{L}\delta\boldsymbol{U}+\delta\boldsymbol{LU}+\delta\boldsymbol{L}\delta\boldsymbol{U})\hat{\boldsymbol{x}}\\&\equiv(\boldsymbol{A}+\delta\boldsymbol{A})\hat{\boldsymbol{x}},\text{其中 }\delta\boldsymbol{A}=-\boldsymbol{E}+\boldsymbol{L}\delta\boldsymbol{U}+\delta\boldsymbol{LU}+\delta\boldsymbol{L}\delta\boldsymbol{U}.\end{aligned}$$

现在合并 $\boldsymbol{E}$、$\delta\boldsymbol{L}$ 和 $\delta\boldsymbol{U}$ 的界并利用三角形不等式界定 $\delta\boldsymbol{A}$：

48

$$
\begin{aligned}
|\delta A| &= |-E + L\delta U + \delta LU + \delta L\delta U| \\
&\leqslant |E| + |L\delta U| + |\delta LU| + |\delta L\delta U| \\
&\leqslant |E| + |L|\cdot|\delta U| + |\delta L|\cdot|U| + |\delta L|\cdot|\delta U| \\
&\leqslant n\varepsilon|L|\cdot|U| + n\varepsilon|L|\cdot|U| + n\varepsilon|L|\cdot|U| + n^2\varepsilon^2|L|\cdot|U| \\
&\approx 3n\varepsilon|L|\cdot|U|.
\end{aligned}
$$

取范数并假定$\| |X| \| = \|X\|$(如前所述对弗罗贝尼乌斯范数、无穷范数和1-范数成立，但对2-范数不成立)，我们得到$\|\delta A\| \leqslant 3n\varepsilon\|L\|\cdot\|U\|$.

因而，为了观察何时高斯消元法是向后稳定的，我们一定要问何时$3n\varepsilon\|L\|\cdot\|U\| = O(\varepsilon)\|A\|$;于是正如我们希望的那样，在扰动理论中$\frac{\|\delta A\|}{\|A\|}$的界将为$O(\varepsilon)$(注意$\delta b = 0$).

几十年的实验经验证实了GEPP几乎总是保持$\|L\|\cdot\|U\| \approx \|A\|$. GEPP保证$L$的每个元素绝对值以1为界，故只要考察$\|U\|$. 定义GEPP的主元增长因子[1](pivot growth factor)为$g_{PP} = \|U\|_{\max}/\|A\|_{\max}$，其中$\|A\|_{\max} = \max_{ij}|a_{ij}|$，故稳定性等价于$g_{PP}$是小的或者作为$n$的函数缓慢地增长. 实际上，$g_{PP}$几乎总是$n$或稍小些. 通常的状态似乎是$n^{2/3}$或者也许恰好为$n^{1/2}$[242](见图2-1). 这就使得对大多数问题选择GEPP算法是可以的. 遗憾的是，在个别例子里，g_{PP}可能达到2^{n-1}.

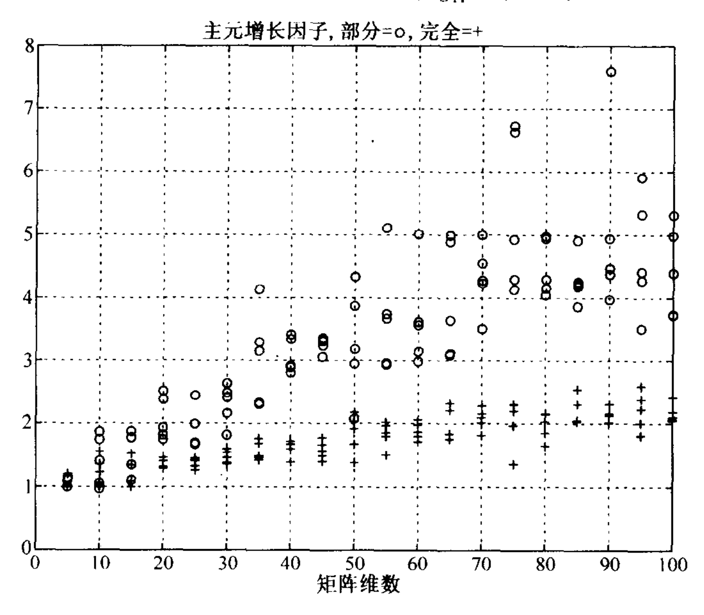


图2-1　随机矩阵的主元增长，$\circ = g_{PP}$，$+ = g_{CP}$

1. 这个定义稍稍不同于通常文献中的定义，但本质上是等价的[121,p.115].

命题2.1 GEPP 保证 $g_{PP} \leq 2^{n-1}$. 这个界是可以达到的.

证明 GEPP 的第一步调整 $\tilde{a}_{jk} = a_{jk} - l_{ji}u_{ik}$,其中 $|l_{ji}| \leq 1$, $|u_{ik}| = |a_{ik}| \leq \max_{rs}|a_{rs}|$, 故 $|\tilde{a}_{jk}| \leq 2 \cdot \max_{rs}|a_{rs}|$. 因而 GEPP 的 $n-1$ 个主步的每一步能使余下的矩阵元素大小加倍,我们得到 2^{n-1} 作为总的界. 观察问题2.14 中的例子,可见这个界是可以达到的. □

把这些界放在一起,因为 $\|\boldsymbol{L}\|_\infty \leq n$, $\|\boldsymbol{U}\|_\infty \leq n g_{PP}\|\boldsymbol{A}\|_\infty$,所以得到

$$\|\delta \boldsymbol{A}\|_\infty \leq 3 g_{PP} n^3 \varepsilon \|\boldsymbol{A}\|_\infty. \tag{2.11}$$

即使 $g_{PP}=1$,在界中的因子 $3g_{PP}n^3$ 使它几乎总是大大地过高估计真正的 $\|\delta \boldsymbol{A}\|$. 例如,若 $\varepsilon = 10^{-7}$ 和 $n=150$, 恰为中等大小的矩阵, 则 $3n^3\varepsilon > 1$, 意味着所有的精度可能丧失. 例2.3 连同真正的向后误差一起用图表示 $3g_{PP}n^3\varepsilon$, 说明结果可能是不好的. $\|\delta \boldsymbol{A}\|$ 通常是 $O(\varepsilon)\|\boldsymbol{A}\|$,故我们可以说 GEPP 实际上是向后稳定的, 即使我们可以构造它失败的例子. 2.4.4 节对 $\boldsymbol{Ax}=\boldsymbol{b}$ 的计算解提供实际的误差界, 这个界比利用 $\|\delta \boldsymbol{A}\|_\infty \leq 3g_{PP}n^3\varepsilon\|\boldsymbol{A}\|_\infty$ 得到的界小很多.

49

可以指出 GECP 甚至比 GEPP 更稳定, 尽管它的主元增长 g_{CP} 满足最差案例的界 [262, p. 213]

$$g_{CP} = \frac{\max_{ij}|u_{ij}|}{\max_{ij}|a_{ij}|} \leq \sqrt{n \cdot 2 \cdot 3^{1/2} \cdot 4^{1/3} \cdots n^{1/(n-1)}} \approx n^{1/2 + \log_e n/4}.$$

这个上界在实践中也是相当大的. g_{CP} 通常的状态是 $n^{1/2}$. 曾有一个众所周知的猜测是 $g_{CP} \leq n$, 但这个猜测最近被证明是错的[99,122]. 因此求 g_{CP} 的一个好的上界仍是个悬而未决的问题. [仍普遍地猜想它为 $O(n)$.]

GECP 用于求主元的额外的 $O(n^3)$ 次比较[GEPP 每步的比较为 $O(n)$ 次, 而 GECP 为 $O(n^2)$ 次]使得 GECP 明显地慢于 GEPP, 特别是在执行浮点运算与做比较一样快的高性能机器上. 所以很少允许使用 GECP(除 2.4.4, 2.5.1 和 5.4.3 节的情形外).

例2.3 图 2-1 和图 2-2 展示了这些向后误差界. 对两个图形, 每个维数生成五个随机矩阵, 它们具有独立正态分布, 均值为 0, 标准差为 1. (测试这种随机矩阵有时可能使人误解实际问题的情况, 但它还是可以提供信息的). 对每个矩阵生成一个类似的随机向量 $\boldsymbol{b}$. 同时使用 GEPP 和 GECP 求解 $\boldsymbol{Ax}=\boldsymbol{b}$. 图 2-1 标出主元增长因子 g_{PP} 和 g_{CP}. 如期望的那样, 在两种情况下它们随着维数增加而缓慢地增长. 图 2-2 指出向后误差的两个上界 $3n^3\varepsilon g_{PP}$(或 $3n^3\varepsilon g_{CP}$)以及 $3n\varepsilon\frac{\||\boldsymbol{L}|\cdot|\boldsymbol{U}|\|_\infty}{\|\boldsymbol{A}\|_\infty}$. 它也指出如定理 2.2 中所述算出的真正的向后误差. 机器精度用一条在 $\varepsilon = 2^{-53} \approx 1.1 \cdot 10^{-16}$ 处的实心的水平线显示. 两个界确实是真正的向后误差界, 但是大好几个数量级. 产生这些图的 Matlab 程序见 HOMEPAGE/Matlab/pivot.m. ◇

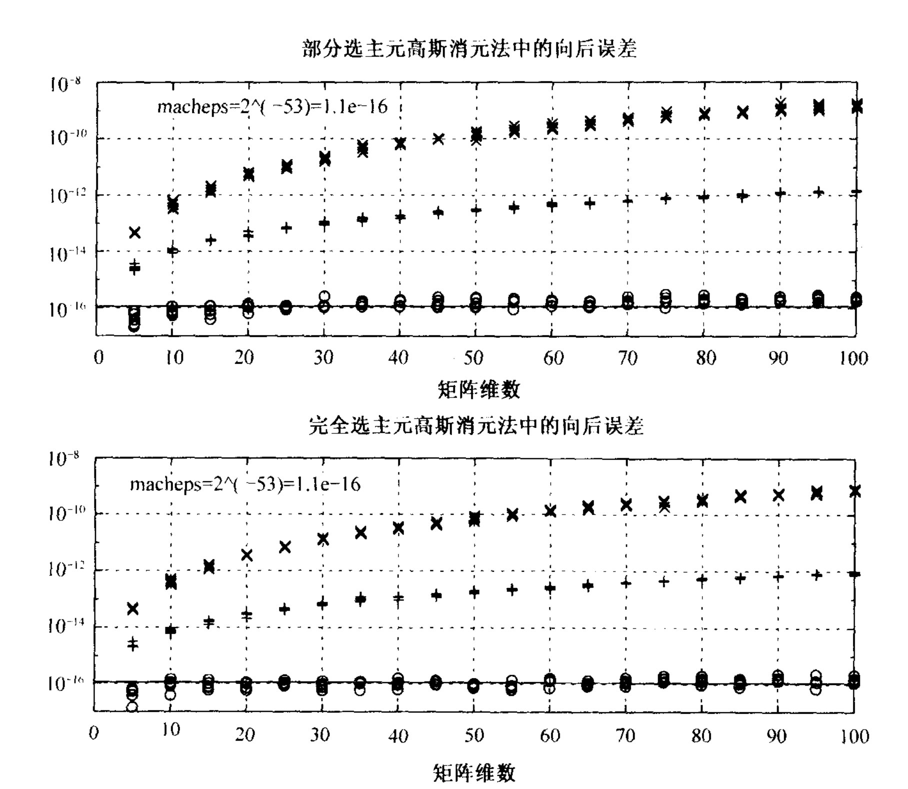


图2-2 关于随机矩阵高斯消元法中的向后误差，$\times=3n^3\varepsilon g$，$+=3n\||L|\cdot|U|\|_\infty/\|A\|_\infty$，$\circ=\|Ax-b\|_\infty/(\|A\|_\infty\|x\|_\infty)$

2.4.3 估计条件数

为了计算以类似(2.5)那样的界为基础的一个实际的误差界，我们需要估计$\|A^{-1}\|$. 因为$\|A\|$容易计算，所以估计条件数$\kappa(A)=\|A^{-1}\|\cdot\|A\|$也就够了. 一个方法是明显地计算$A^{-1}$并计算它们的范数. 然而，这将花费$2n^3$，大于高斯消元法原有的$\frac{2}{3}n^3$.（这就推出通过计算$A^{-1}$,然后用$b$相乘来求解$Ax=b$是没有价值的. 即使有许多不同的$b$，这个结论也是成立的，见问题2.2.）如果计算误差界代价昂贵，则大多数用户不愿意费心计算它们. 我们另辟蹊径，不去计算A^{-1}，而是设计一个便宜得多的算法估计$\|A^{-1}\|$. 这样的一个算法称为条件估计量，并且应该有下列性质：

50

1. 给出A的L和U因子，它应花费$O(n^2)$,对足够大的n，它与GEPP的$\frac{2}{3}n^3$代价相比较是可以忽略不计的.

2. 它应当提供一个估计，它几乎总是不超过$\|A^{-1}\|$的10倍. 这是所有的人需要的误差界，它告诉你大约有多少精确的十进位数字(10倍误差是一个十进位数字)[1].

有许多这样的估计量可利用(见[146]综述). 我们选择介绍一个除解$Ax=b$之外，还可广泛应用的估计量，所花费的时间稍慢于$Ax=b$的特定的算法(但仍相当快). 像其他大多数算法一样我们的估计量保证只产生$\|A^{-1}\|$的一个下界，而不是上界. 经验表明，它几乎总是不超过10倍$\|A^{-1}\|$的一个因子，通常是$\|A^{-1}\|$的2到3倍. 对图2-1和图2-2中的矩阵，条件数从10变到10^5，估计量等于条件数的83%. 而在最差情况估计量等于条件数的0.43倍. 这足以精确估计最后的解答中准确的十进制数位.

倘若对任意的$\boldsymbol{x}$和$\boldsymbol{y}$可计算$\boldsymbol{Bx}$和$\boldsymbol{B}^{\mathrm{T}}\boldsymbol{y}$,算法估计矩阵$\boldsymbol{B}$的1-范数$\|\boldsymbol{B}\|_1$. 将对$\boldsymbol{B}=\boldsymbol{A}^{-1}$应用算法，故我们需要计算$\boldsymbol{A}^{-1}\boldsymbol{x}$和$\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{y}$,即解线性方程组. 其代价正好是$\boldsymbol{A}$的$\boldsymbol{LU}$分解给出的$O(n^2)$. 算法在[138,146,148]中讨论，最新的版本在[147]中. 记住$\|\boldsymbol{B}\|_1$由下式定义.

$$\|\boldsymbol{B}\|_1 = \max_{x\neq 0}\frac{\|\boldsymbol{Bx}\|_1}{\|\boldsymbol{x}\|_1} = \max_j \sum_{i=1}^{n}|b_{ij}|.$$

容易证明满足$\boldsymbol{x}\neq 0$时最大值是在$\boldsymbol{x}=\boldsymbol{e}_{j_0}=[0,\cdots,0,1,0,\cdots,0]^{\mathrm{T}}$(单个非零元是分量$j_0$，其中$\max_j\sum_i|b_{ij}|$在$j=j_0$处出现)处达到.

在遍历所有的$e_j(j=1,\cdots,n)$上搜索，意味着计算$\boldsymbol{B}=\boldsymbol{A}^{-1}$的所有列，这样代价太昂贵. 相反，因为$\|\boldsymbol{B}\|_1=\max_{\|\boldsymbol{x}\|_1\leq 1}\|\boldsymbol{Bx}\|_1$,所以可使用集合$\|\boldsymbol{x}\|_1\leq 1$之内关于$f(\boldsymbol{x})\equiv\|\boldsymbol{Bx}\|_1$的斜坡攀登(hill climbing)或梯度上升(gradient ascent). $\|\boldsymbol{x}\|_1\leq 1$显然是向量的一个凸集，而$f(\boldsymbol{x})$是凸函数，因为$0\leq\alpha\leq 1$推出$f(\alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y})=\|\alpha\boldsymbol{Bx}+(1-\alpha)\boldsymbol{By}\|_1\leq\alpha\|\boldsymbol{Bx}\|_1+(1-\alpha)\|\boldsymbol{By}\|_1=\alpha f(\boldsymbol{x})+(1-\alpha)f(\boldsymbol{y})$.

沿梯度上升方向来极大化$f(\boldsymbol{x})$意味着只要$f(\boldsymbol{x})$增加，在梯度$\nabla f(\boldsymbol{x})$(如果它存在)方向移动$\boldsymbol{x}$. $f(\boldsymbol{x})$的凸性意味着$f(\boldsymbol{y})\geq f(\boldsymbol{x})+\nabla f(\boldsymbol{x})\cdot(\boldsymbol{y}-\boldsymbol{x})$[若$\nabla f(\boldsymbol{x})$存在]. 为计算$\nabla f$,假定在$f(\boldsymbol{x})=\sum_i\left|\sum_j b_{ij}x_j\right|$中所有$\sum_j b_{ij}x_j\neq 0$(这几乎总是成立的). 设$\zeta_i=\mathrm{sign}(\sum_j b_{ij}x_j)$,因而$\zeta_i=\pm 1$并且$f(\boldsymbol{x})=\sum_i\sum_j\zeta_i b_{ij}x_j$. 于是$\frac{\partial f}{\partial x_k}=\sum_i\zeta_i b_{ik}$并且$\nabla f=\boldsymbol{\zeta}^{\mathrm{T}}\boldsymbol{B}=(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\zeta})^{\mathrm{T}}$.

51~52

总之，为计算$\nabla f(\boldsymbol{x})$使用三个步骤：$w=\boldsymbol{Bx}$，$\boldsymbol{\zeta}=\mathrm{sign}(w)$，和$\nabla f=\boldsymbol{\zeta}^{\mathrm{T}}\boldsymbol{B}$.

---

1. 正如早先所述的，人们从未找到一个具有某种保证的精度而且它比明显地计算$\boldsymbol{A}^{-1}$的代价足够便宜的逼近$\|\boldsymbol{A}^{-1}\|$的估计量. 曾猜测不存在这样的估计量，但这个猜测未被证明.

算法 2.5　Hager 的条件估计量获得 $\|B\|_1$ 的一个下界 $\|w\|_1$

```
任选 x 满足 ||x||_1 = 1                 /* e.g. x_i = 1/n */
repeat
  w = Bx, ζ = sign(w), z = B^T ζ       /* z^T = ∇f */
  if ||z||_∞ ≤ z^T x then
    return ||w||_1
  else
    x = e_j where |z_j| = ||z||_∞
  end if
end repeat
```

定理 2.6　1. 当获得 $\|w\|_1$ 时，$\|w\|_1=\|Bx\|_1$ 是 $\|Bx\|_1$ 的一个局部极大值.

2. 否则，$\|Be_j\|$(在循环的末端) > $\|Bx\|$(在循环的开始)，因而算法在极大化 $f(x)$ 方面已获得进展.

证明

(1)此时 $\|z\|_\infty \leqslant z^Tx$. 在 x 的附近，$f(x)=\|Bx\|_1=\sum_i\sum_j\zeta_i b_{ij}x_j$ 对于 x 是线性的，因而 $f(y)=f(x)+\nabla f(x)\cdot(y-x)=f(x)+z^T(y-x)$，其中 $z^T=\nabla f(x)$. 为证明 x 是一个局部极大值点，希望当 $\|y_1\|=1$ 时 $z^T(y-x)\leqslant 0$. 计算

$$z^T(y-x)=z^Ty-z^Tx=\sum_i z_i\cdot y_i-z^Tx\leqslant\sum_i|z_i|\cdot|y_i|-z^Tx$$
$$\leqslant\|z\|_\infty\cdot\|y\|_1-z^Tx=\|z\|_\infty-z^Tx\leqslant 0 \text{ 正如希望的那样.}$$

(2)此时 $\|z\|_\infty>z^Tx$. 选择 $\tilde{x}=e_j\cdot\mathrm{sign}(z_j)$，其中选择 j 使得 $|z_j|=\|z\|_\infty$. 于是

$$f(\tilde{x})\geqslant f(x)+\nabla f\cdot(\tilde{x}-x)=f(x)+z^T(\tilde{x}-x)$$
$$=f(x)+z^T\tilde{x}-z^Tx=f(x)+|z_j|-z^Tx>f(x),$$

其中最后的不等式由构造知道是成立的.　□

[53] Higham[147,148]尝试用大小为 10,25,50 和条件数为 $\kappa=10,10^3,10^6,10^9$ 的许多随机矩阵测试了这个算法的一个稍稍改进的版本. 在最差的情况计算的 κ 低估真正的 κ(相差一个因子 0.44). 算法可以利用 LAPACK 中的子程序 slacon. LAPACK 程序 sgesvx 在内部称为 slacon 并且获得估计的条件数(它们实际上获得估计的条件数的倒数，以免对精确的奇异阵上溢). 一个不同的条件估计量可以利用 Matlab 中的 rcond. Matlab 程序 cond 利用 5.4 节中讨论的算法计算精确的条件数 $\|A^{-1}\|_2\|A\|_2$. 它比 rcond 花费更加高的代价.

估计相对条件数

我们也可以利用上段中的算法由界(2.8)或由(2.9)计算界 $\|\,|A^{-1}|\cdot|r|\,\|_\infty$ 来估

计相对条件数 $\kappa_{CR}(A)=\| |A^{-1}|\cdot|A| \|_\infty$. 可以同时化为相同的估计 $\| |A^{-1}|\cdot g\|_\infty$ 的问题，其中 g 是一个非负元向量. 为看看为什么这样，设 e 是一个全 1 的向量. 从引理 1.7 的第 5 部分知，当矩阵 X 有非负元时，$\|X\|_\infty=\|Xe\|_\infty$. 于是

$$\| |A^{-1}|\cdot|A| \|_\infty=\| |A^{-1}|\cdot|A|e\|_\infty=\| |A^{-1}|\cdot g\|_\infty, \text{其中 } g=|A|e.$$

下面我们来估计 $\| |A^{-1}|\cdot g\|_\infty$. 设 $G=\mathrm{diag}(g_1,\cdots,g_n)$；则 $g=Ge$. 因而

$$\begin{aligned}\| |A^{-1}|\cdot g\|_\infty &= \| |A^{-1}|\cdot Ge\|_\infty=\| |A^{-1}|\cdot G\|_\infty \\ &= \| |A^{-1}G| \|_\infty=\|A^{-1}G\|_\infty. \end{aligned} \tag{2.12}$$

因为对任意的矩阵 Y, $\|Y\|_\infty=\| |Y| \|_\infty$, 所以最后的不等式成立. 因而，它足够估计矩阵 $A^{-1}G$ 的无穷范数. 我们可以应用 Hager 算法(算法 2.5)来做这件事，对矩阵 $(A^{-1}G)^T=GA^{-T}$, 估计 $\|(A^{-1}G)^T\|_1=\|A^{-1}G\|_\infty$ (见引理 1.7 第 6 部分). 这要求矩阵 GA^{-T} 和它的转置 $A^{-1}G$ 相乘. 用 G 相乘是容易的，因为它是对角阵，而用 A^{-1} 和 A^{-T} 相乘，正如上节中我们所做的那样，要利用 A 的 LU 分解.

2.4.4 实际的误差界

对 $Ax=b$ 的近似解 $\hat{x}$ 提出两个实际的误差界. 第一个界利用不等式(2.5)得到

$$\text{error}=\frac{\|\hat{x}-x\|_\infty}{\|\hat{x}\|_\infty}\leqslant\|A^{-1}\|_\infty\cdot\frac{\|r\|_\infty}{\|\hat{x}\|_\infty}, \tag{2.13}$$

其中 $r=A\hat{x}-b$ 是残差. 对 $B=A^{-T}$ 应用算法 2.5 估计 $\|A^{-1}\|_\infty$, 估计 $\|B\|_1=\|A^{-T}\|_1=\|A^{-1}\|_\infty$ (见引理 1.7 的第 5 和第 6 部分).

第二个误差界来自紧凑的不等式(2.9)：

$$\text{error}=\frac{\|\hat{x}-x\|_\infty}{\|\hat{x}\|_\infty}\leqslant\frac{\| |A^{-1}|\cdot|r| \|_\infty}{\|\hat{x}\|_\infty}. \tag{2.14}$$

54

利用基于(2.12)式的算法估计 $\| |A^{-1}|\cdot|r| \|_\infty$. 误差界(2.14)(正如下面小节“可能导致什么故障”中所述的修正)用 LAPACK 程序 sgesvx 来计算. 误差界的 LAPACK 变量名是 FERR，它是向前误差(Forward ERRor)的缩写.

例 2.4 对图 2-1 和图 2-2 中同样的一组例子已算出第一个误差界和真正的误差界，在图 2-3 中标出结果的位置. 对每个用 GEPP 求解的问题 $Ax=b$ 在点(真正的误差，误差界)上标一个 ∘，而对每个用 GECP 求解的问题 $Ax=b$ 在点(真正的误差，误差界)上标一个 +. 如果误差界等于真正的误差，则 ∘ 或 + 位于对角实线上. 因为误差界总是超过真正的误差，所以这些 ∘ 和 + 位于这条对角线上方. 当误差界是介于真正误差值与其 10 倍值之间时，∘ 或者 + 出现在对角实线和第一条上对角虚线之间. 当误差界是介于真正误差值的 10 倍与 100 倍之间时，∘ 和 + 出现在前两条上对角虚线之间. 大多数误差界是在这个范围内，少数误差界有 1000 倍真正的误差那样大. 因此，计算的误差界低估解答中正确的小数位数 1 位或 2 位，在罕见的情况差不多 3 位. 制作这个图表的 Matlab 代码与前面一样，是 HOMEPAGE/Matlab/pivot.m. ◇

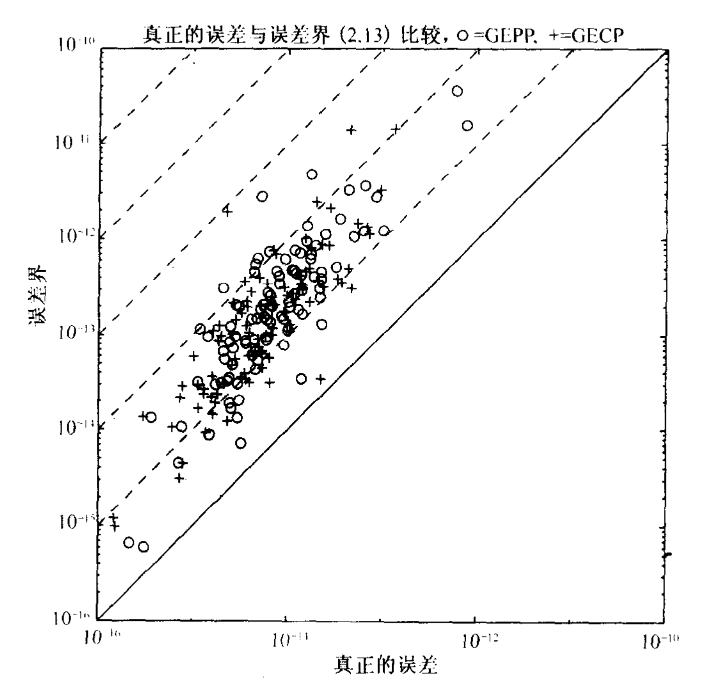


图2-3　画出误差界(2.13)与真正的误差比较，o=GEPP，+=GECP

例2.5　我们提出一个例子,说明两个误差界(2.13)和(2.14)之间的差别. 此例也说明有时GECP可能比GEPP更精确. 选择一组如下构造坏的比例的例子. 每个测试矩阵具有形式$A=DB$，维数从5到100. B等于一个单位阵加非常小的大约10^{-7}的随机非对角元，故它是非常良态的. D是一个具有从1到10^{14}几何比例元素的对角阵. (换言之，$d_{i+1,i+1}/d_{ii}$对一切i是相同的.) 矩阵A的条件数$\kappa(A)=\|A^{-1}\|_\infty\cdot\|A\|_\infty$几乎等于$10^{14}$，这是非常病态的，虽然它们的相对条件数$\kappa_{CR}(A)=\||A^{-1}|\cdot|A|\|_\infty=\||B^{-1}|\cdot|B|\|_\infty$几乎都为1. 如前述一样，机器精度是$\varepsilon=2^{-53}\approx10^{-16}$. 这些例子是用同样的Matlab代码HOMEPAGE/Matlab/pivot.m计算的.

对任何例子主元增长因子g_{PP}和g_{CP}决不会大于1.33左右，并且在任何情况下由定理2.2给出的向后误差决不会超过10^{-15}. Hager的估计量在所有的情况下都是非常精确的，获得真正的条件数10^{14}达到许多小数位.

图2-4对这些例子标出误差界(2.13)和(2.14)，以及定理2.3中公式给出的按分量的相对误差界.

图2-4a左上角中成串的加号说明当GECP计算具有靠近10^{-15}的一个微小误差的解答时，误差界(2.13)通常接近于10^{-2}，这个界是十分不利的. 这是因为条件数是10^{14}，故除非向后误差比$\varepsilon\approx10^{-16}$小很多，而这是不太可能的，所以误差界将接近于$10^{-16}\cdot10^{14}=10^{-2}$. 同一图形中上部中成串的圆圈说明当误差界(2.13)再一次接近于

通常的 10^{-2}时，GEPP 得到一个较大的约为 10^{-8}误差.

相反地，正如图 2-4b 中对角线上的加号和圆圈所说明的那样，误差界(2.14)几乎是完全精确的. 这个图表再次说明 GECP 几乎是完全精确的，而 GEPP 丧失大约一半的精度. 这个精度差别用图 2-4c 来说明. 它指出来自定理 2.3 的关于 GEPP 和 GECP 的按分量的相对向后误差，这个图表使它明显可以看出按分量相对意义下 GECP 有几乎正确的向后误差，因为对应的按分量的相对条件数是 1，所以精度是完美的. 另一方面，GEPP 在丧失 5 到 10 个十进位数字这种意义下不是完全稳定的.

55 ~ 56

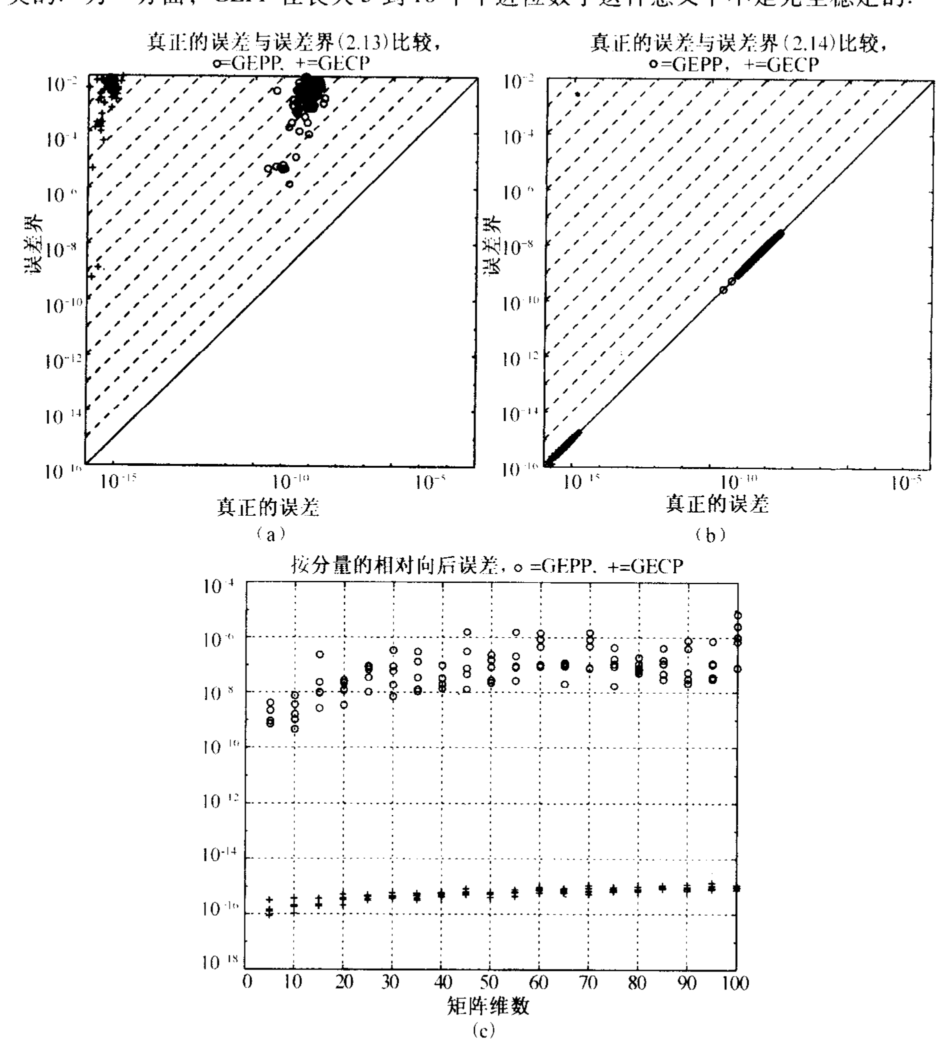


图 2-4 (a) 标出误差界(2.13)与真正的误差的位置；(b) 标出误差界(2.14)与真正的误差的位置；(c) 标出来自定理 2.3 的按分量的相对向后误差

在2.5节中将说明如何迭代改进计算解$\hat{x}$. 这个方法的单步将使得用GECP计算的解像用GECP得到的解那样准确. 因为GECP在实际上比GEPP花费更大的代价, 所以它非常难得使用. ◇

可能导致什么故障

遗憾的是, 正如2.4节开始提到的那样, 实际操作时误差界(2.13)和(2.14)不保证在所有情况提供严密的界. 在本节中我们描述它们可能失效的(罕见的!)方法以及在实际上使用的部分补救办法.

首先, 如2.4.3节中所述, 从算法2.5(或类似的算法)估计$\|A^{-1}\|$只提供一个下界, 虽然它小于超过10倍的可能性非常低.

其次, 有一个小的但不可忽视的可能性, 即在$r=A\hat{x}-b$求值中的舍入可能使$\|r\|$人为地小, 实际上为0, 并且也使得计算的误差界非常小. 考虑到这个可能性, 人们可以在$|r|$上增加一个小量去说明它: 从问题1.10知道计算r中的舍入被界定为

$$|(A\hat{x}-b)-\mathrm{fl}(A\hat{x}-b)|\leqslant(n+1)\varepsilon(|A|\cdot|\hat{x}|+|b|) \tag{2.15}$$

57 ~ 58

故我们可以在界(2.14)中用$|r|+(n+1)\varepsilon(|A|\cdot|\hat{x}|+|b|)$代替$|r|$(这是用LAPACK代码sgesvx做的)或者在界(2.13)中用$\|r\|+(n+1)\varepsilon(\|A\|\cdot\|\hat{x}\|+\|b\|)$代替$\|r\|$. 因子$n+1$通常是相当大的, 如果必要的话可以略去.

最后, 在对非常病态矩阵执行高斯消元法的过程中, 舍入能产生不准确的L和U以致界(2.14)非常低.

例2.6 我们介绍一个由W. Kahan发现的例子, 它说明得出真正有保证的误差界的困难. 此例中矩阵A是恰好奇异的. 所以$\frac{\|x-\hat{x}\|}{\|x\|}$的任何误差界应该是1或较大的数, 来指出计算解中没有数字是正确的, 因为真正的解不存在.

高斯消元法中的舍入误差将得到非奇异的但非常病态的因子L和U. 就这个例子而言, 利用有IEEE双精度运算的Matlab计算, 因为舍入计算的残差r结果变成精确的0, 因而两个误差界(2.13)和(2.14)同时为0. 若通过加上$4\varepsilon(\|A\|\cdot\|\hat{x}\|+\|b\|)$来补救界(2.13), 则它将如要求的那样大于1.

遗憾的是, 第二个"紧凑的"误差界(2.14)大约是10^{-7}, 它错误地指示计算解的七个数字是正确的.

下面来构造这样的例子. 设$\chi=3/2^{29}$, $\zeta=2^{14}$,

$$A=\begin{bmatrix}\chi\cdot\zeta & -\zeta & \zeta\\ \zeta^{-1} & \zeta^{-1} & 0\\ \zeta^{-1} & -\chi\cdot\zeta^{-1} & \zeta^{-1}\end{bmatrix}\approx\begin{bmatrix}9.1553\cdot10^{-5} & -1.6384\cdot10^{4} & 1.6384\cdot10^{4}\\ 6.1035\cdot10^{-5} & 6.1035\cdot10^{-5} & 0\\ 6.1035\cdot10^{-5} & -3.4106\cdot10^{-13} & 6.1035\cdot10^{-5}\end{bmatrix},$$

而$b=A\cdot[1,1+\varepsilon,1]^{\mathrm{T}}$. 可以没有任何舍入误差算出$A$, 但$b$有一点舍入, 这意味着它在$A$的列张成的空间中不是精确的, 故$Ax=b$无解. 执行高斯消元法, 得到

$$L\approx\begin{bmatrix}1 & 0 & 0\\ 0.66666 & 1 & 0\\ 0.66666 & 1.0000 & 1\end{bmatrix}$$

和

$$U \approx \begin{bmatrix} 9.1553 \cdot 10^{-5} & -1.6384 \cdot 10^{4} & 1.6384 \cdot 10^{4} \\ 0 & 1.0923 \cdot 10^{4} & -1.0923 \cdot 10^{4} \\ 0 & 0 & 1.8190 \cdot 10^{-12} \end{bmatrix},$$

得到 A^{-1} 的计算值为

$$A^{-1} \approx \begin{bmatrix} 2.0480 \cdot 10^{3} & 5.4976 \cdot 10^{11} & -5.4976 \cdot 10^{11} \\ -2.0480 \cdot 10^{3} & -5.4976 \cdot 10^{11} & 5.4976 \cdot 10^{11} \\ -2.0480 \cdot 10^{3} & -5.4976 \cdot 10^{11} & 5.4976 \cdot 10^{11} \end{bmatrix}$$

这意味着 $|A^{-1}| \cdot |A|$ 计算值的所有元近似地等于 $6.7109 \cdot 10^{7}$，故算出 $\kappa_{CR}(A)$ 为 $O(10^{7})$. 换言之，误差界指出计算解的大约 $16-7=9$ 个数字是正确的，然而却没有一个数字是正确的. [59]

除非大的主元增长，我们可以证明不可能通过这里说明的现象人为地作出小的界(2.13)(用适当地增大的 $\|r\|$).

类似地，Kahan 已找出一族 $n\times n$ 阶奇异阵，把一个微小的元(2^{-n} 左右)改变为零，把 $\kappa_{CR}(A)$ 降低为 $O(n^{3})$. 可以类似地构造 A 不是恰好奇异的例子，使得按精确的算术运算界(2.13)和界(2.14)是正确的，但是舍入使得它们过分地小. ◇

2.5 改进解的精度

我们刚刚已经看到求解 $Ax=b$ 中的误差可能如 $\kappa(A)\varepsilon$ 那样大. 若这个误差太大，我们能做些什么事情呢？一个可能性是以高精度再执行整个计算，但这个做法在时间和空间方面的代价可能是十分昂贵的. 幸运的是，只要 $\kappa(A)$ 不是太大时，为得到一个更为准确的解有许多更加廉价的方法.

为求解任何方程 $f(x)=0$，我们可尝试用牛顿法改进近似解 x_i 得到 $x_{i+1}=x_i-\frac{f(x_i)}{f'(x_i)}$. 对 $f(x)=Ax-b$ 应用这个方法得到迭代精化的单步

$$r = Ax_i - b$$
解 $Ad = r$ 得 d
$$x_{i+1} = x_i - d$$

若可以精确地计算 $r=Ax_i-b$ 和精确地求解 $Ad=r$，则我们将在一步中完成，这就是把牛顿法应用于一个线性问题所期望的事. 舍入误差妨碍这个直接的收敛性. 当 A 是如此病态以致解 $Ad=r$(及 $Ax_0=b$)是相当不准确时，此算法是令人感兴趣的，并且正好是有用的.

定理 2.7 假定 r 按双精度计算，$\kappa(A)\cdot\varepsilon<c\equiv\frac{1}{3n^{3}g+1}<1$，其中 n 是 A 的维数，g

是主元增长因子. 则反复的迭代精化以

$$\frac{\|x_i - A^{-1}b\|_\infty}{\|A^{-1}b\|_\infty} = O(\varepsilon)$$

收敛.

注意条件数在最后的误差界中不出现. 这意味着倘若 $\kappa(A)\varepsilon$ 充分小于1时, 人们能精确计算解答, 而与条件数无关. (实际上 c 是一个过分保守的上界, 即使当 $\kappa(A)\varepsilon$ 大于 c 时算法通常也能取得成功.)

60

证明的梗概　为了使证明保持清晰, 仅仅考虑最重要的舍入误差. 为简洁起见, 把 $\|\cdot\|_\infty$ 简化为 $\|\cdot\|$. 我们的目标是证明

$$\|x_{i+1} - x\| \leqslant \frac{\kappa(A)\varepsilon}{c}\|x_i - x\| \equiv \zeta\|x_i - x\|.$$

由设 $\zeta < 1$, 故此不等式推出误差 $\|x_{i+1} - x\|$ 单调地递减到零. (实际上自始至终它不递减到零, 因为我们忽略赋值 $x_{i+1} = x_i - d$ 中的舍入误差.)

通过估计计算的残差 r 中的误差入手, 得到 $r = \mathrm{fl}(Ax_i - b) = Ax_i - b + f$, 由问题1.10的结果 $|f| \leqslant n\varepsilon^2(|A|\cdot|x_i| + |b|) + \varepsilon|Ax_i - b| \approx \varepsilon|Ax_i - b|$. ε^2 项来自 r 的双精度计算, ε 项来自双精度的结果舍入到单精度. 因为 $\varepsilon^2 \ll \varepsilon$, 所以在 $|f|$ 的界中忽略 ε^2 项.

下面我们得到 $(A + \delta A)d = r$, 从界(2.11)知道 $\|\delta A\| \leqslant \gamma\cdot\varepsilon\cdot\|A\|$, 其中 $\gamma = 3n^3g$, 虽然这个值通常是过分大了. 如前面所提及的那样, 我们可通过假定 $x_{i+1} = x_i - d$ 精确成立来简化问题.

继续忽略所有的 ε^2 项, 得到

$$\begin{aligned}
d &= (A + \delta A)^{-1}r = (I + A^{-1}\delta A)^{-1}A^{-1}r \\
&= (I + A^{-1}\delta A)^{-1}A^{-1}(Ax_i - b + f) \\
&= (I + A^{-1}\delta A)^{-1}(x_i - x + A^{-1}f) \\
&\approx (I - A^{-1}\delta A)(x_i - x + A^{-1}f) \\
&\approx x_i - x - A^{-1}\delta A(x_i - x) + A^{-1}f.
\end{aligned}$$

所以 $x_{i+1} - x = x_i - d - x = A^{-1}\delta A(x_i - x) - A^{-1}f$, 故

$$\begin{aligned}
\|x_{i+1} - x\| &\leqslant \|A^{-1}\delta A(x_i - x)\| + \|A^{-1}f\| \\
&\leqslant \|A^{-1}\|\cdot\|\delta A\|\cdot\|x_i - x\| + \|A^{-1}\|\cdot\varepsilon\cdot\|Ax_i - b\| \\
&\leqslant \|A^{-1}\|\cdot\|\delta A\|\cdot\|x_i - x\| + \|A^{-1}\|\cdot\varepsilon\cdot\|A(x_i - x)\| \\
&\leqslant \|A^{-1}\|\cdot\gamma\varepsilon\cdot\|A\|\cdot\|x_i - x\| + \|A^{-1}\|\cdot\|A\|\cdot\varepsilon\cdot\|x_i - x\| \\
&= \|A^{-1}\|\cdot\|A\|\cdot\varepsilon\cdot(\gamma + 1)\cdot\|x_i - x\|,
\end{aligned}$$

故若

$$\zeta = \|A^{-1}\|\cdot\|A\|\cdot\varepsilon(\gamma + 1) = \kappa(A)\varepsilon/c < 1,$$

则迭代收敛. □

61

迭代精化(或其他牛顿法的变形)也可用于改进许多其他线性代数问题的精度.

2.5.1 单精度迭代精化

第一次阅读可以跳过本节.

有时不能利用双精度进行迭代精化. 例如，若输入数据已经是双精度，则我们需要以四倍精度计算残差 $\boldsymbol{r}$，而这可能是达不到的. 在某些机器上，像英特尔奔腾那样，可利用双倍扩展精度，它比双精度多提供 11 个小数位(见 1.5 节). 这虽然不像四倍精度(它至少需要 $2\times53=106$ 个小数位)那样精确，但仍旧显著地改进了精确性.

但是如果这些选项没有一个可利用的话，当以单精度(即如输入数据那样相同的精度)计算残差 $\boldsymbol{r}$ 时仍旧可以进行迭代精化. 此时，定理 2.7 更不会成立. 另一方面，下列定理指出在一定的技术假设之下，以单精度作一步迭代精化仍旧是值得的，因为它把定理 2.3 中定义的按分量的相对向后误差减少至 $O(\varepsilon)$. 如果来自 2.2.1 节的相应的相对条件数 $\kappa_{CR}(\boldsymbol{A})=\||\boldsymbol{A}^{-1}|\cdot|\boldsymbol{A}|\|_\infty$ 比通常的条件数 $\kappa(\boldsymbol{A})=\|\boldsymbol{A}^{-1}\|_\infty\cdot\|\boldsymbol{A}\|_\infty$ 小得多，则解答也将更加准确.

定理 2.8 *假定 $\boldsymbol{r}$ 以单精度计算且*

$$\|\boldsymbol{A}^{-1}\|_\infty\cdot\|\boldsymbol{A}\|_\infty\cdot\frac{\max_i(|\boldsymbol{A}|\cdot|\boldsymbol{x}|)_i}{\min_i(|\boldsymbol{A}|\cdot|\boldsymbol{x}|)_i}\cdot\varepsilon<1.$$

则一步迭代精化得到 $\boldsymbol{x}_1$ 使得 $(\boldsymbol{A}+\delta\boldsymbol{A})\boldsymbol{x}_1=\boldsymbol{b}+\delta\boldsymbol{b}$，$|\delta a_{ij}|=O(\varepsilon)|a_{ij}|$，$|\delta b_i|=O(\varepsilon)|b_i|$. 换言之，按分量的相对向后误差尽可能小. 例如，这意味着若 $\boldsymbol{A}$ 和 b 是稀疏的，则 $\delta\boldsymbol{A}$ 和 $\delta\boldsymbol{b}$ 分别有 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 相同的稀疏性结构.

证明可见[149]，更多的细节可见[14,225,226,227].

单精度迭代精化和误差界(2.14)是用 LAPACK 中像 sgesvx 那样的程序实现的.

例 2.7 考察例 2.5 中相同的矩阵，当计算中止时($\varepsilon\approx10^{-6}$)以相同的精度执行迭代精化的一步. 对这些例子，通常的条件数是 $\kappa(\boldsymbol{A})\approx10^{14}$，而 $\kappa_{CR}(\boldsymbol{A})\approx1$，因而我们期望一个大的精度改进. 确实，GEPP 按分量的相对误差被压低到 10^{-15} 以下，而对应的来自(2.14)的误差也被压低到 10^{-15} 以下. 这个例子的 Matlab 代码是 HOMEPAGE/Matlab/pivot. m. ◇

2.5.2 平衡

改进解线性方程组中的误差有一个更普遍的技巧：平衡. 这需要选择一个适当的对角阵 $\boldsymbol{D}$ 并求解 $\boldsymbol{DAx}=\boldsymbol{Db}$ 来代替 $\boldsymbol{Ax}=\boldsymbol{b}$. 尝试选择 $\boldsymbol{D}$ 使 $\boldsymbol{DA}$ 的条件数小于 $\boldsymbol{A}$ 的条件数. 例如在例 2.7 中选择 d_{ii} 是 $\boldsymbol{A}$ 的 i 行的 2 范数的倒数，使得 $\boldsymbol{DA}$ 几乎等于对角阵，把它的条件数从 10^{14} 减少到 1. 可以证明这个方法选择的 $\boldsymbol{D}$ 减少 $\boldsymbol{DA}$ 的条件数不超过任意对角阵 $\boldsymbol{D}$ 的最小可能值的 $\sqrt{n}$ 倍[244]. 实际上也可选择两个对角阵 $\boldsymbol{D}_{row}$ 和 $\boldsymbol{D}_{col}$ 和解 $(\boldsymbol{D}_{row}\boldsymbol{AD}_{col})\bar{x}=\boldsymbol{D}_{row}\boldsymbol{b}$，$x=\boldsymbol{D}_{col}\bar{x}$. [62]

迭代精化和平衡的技巧是分别用 LAPACK 中像 sgerfs 和 sgeequ 这样的子程序

来实现的. 这些子程序通过驱动程序 sgesvx 轮流使用.

2.6 高性能分块算法

在2.3节末尾，我们讨论了改变算法2.2的高斯消元法执行中的三个嵌套循环的次序可以改变执行速度的数量级，这与所用的计算机以及求解的问题有关. 本节中将探索这种情形的理由，并介绍一些考虑到这些原因而仔细编写的线性代数软件. 这些软件使用所谓的分块算法，因为它们对矩阵的方的或长方的子块在它们最内层循环上运算而不是在整个行或列上. 这些代码可在 LAPACK(用 Fortran 语言，在 NETLIB/lapack 中)[1]和 ScaLAPACK(在 NETLIB/scalapack 中)那样的不受版权限制的软件库中得到. LAPACK(以及它的其他语言版本)适用于个人计算机、工作站、向量计算机以及共享存储器的并行计算机. 这些计算机包含 Sun SPARC center 2000[238]，SGI Power Challenge[223]，DEC AlphaServer 8400[103]，和 Cray C90/J90[253,254]. ScaLAPACK 适用于像 IBM SP-2[256]，Intel Paragon[257]，Cray T3 系列[255]以及工作站网络[9]那样的分布式存储器并行计算机. 这些程序库及其综合手册在 NETLIB 上可以得到[10,34].

关于高性能(特别是并行的)计算机算法更全面的讨论可以在网址 PARALLELHOMEPAGE 上找到.

LAPACK 最初是由于它的前身 LINPACK 和 EISPACK(在 NETLIB 上也可得到)在某些高性能机器上糟糕的性能所促成的. 例如，考虑下面的表格，它给出是一台1980年代晚期的超级计算机 Cray YMP 上 LINPACK 的楚列斯基程序 Spofa 的速度(按 Mflops 度量)，楚列斯基是适用于对称正定矩阵的高斯消元法的一个变形. 我们在2.7节中再深入地讨论；这里知道它非常类似于算法2.2就足够了. 表格也包括几个其他线性代数运算的速度. Cray YMP 是一台可以同时使用最多8个处理器的并行计算机，因而表中一列是1个处理器的数据，另一列是所有8个处理器都使用时的数据.

63

	1个处理器	8个处理器
最大速度	330	2640
矩阵-矩阵相乘 ($n=500$)	312	2425
矩阵-向量相乘 ($n=500$)	311	2285
解 $TX=B$ ($n=500$)	309	2398
解 $Tx=b$ ($n=500$)	272	584
LINPACK (楚列斯基, $n=500$)	72	72
LAPACK (楚列斯基, $n=500$)	290	1414
LAPACK (楚列斯基, $n=1000$)	301	2115

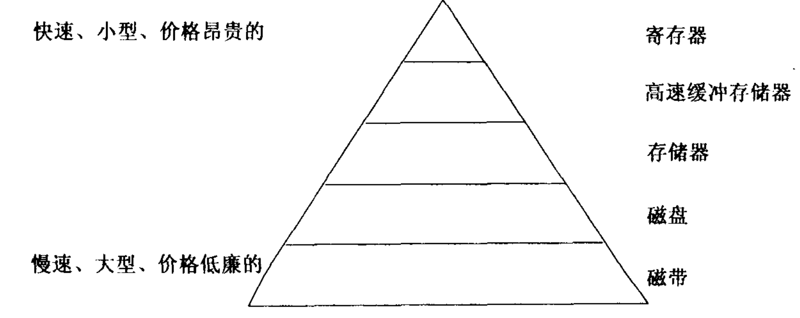
1. 也可利用 LAPACK 中的 C 版本，称为 CLAPACK. LAPACK++ (在 NETLIB/c++/lapack++ 上)和 LAPACK90(在 NETLIB/lapack 90 上)分别是 LAPACK 的 C++ 和 Fortran 90 接口.

首行上机器的最大速度是下面的各行所对应数字的上界. 下面的四行上的基本线性代数运算已经在 Cray YMP 上用特别设计的子程序快速测量过. 除了解单个三角形方程组 $Tx=b$ 之外, 因它未有效地使用 8 个处理器. 它们全都得到接近于最大可能的速度. 解 $TX=B$ 与解多个右端项的三角形方程组有关(B 是一个方阵). 这些数是对大的矩阵和向量的($n=500$).

第 6 行中来自 LINPACK 的楚列斯基程序的执行比其他的运算慢得多, 虽然所操作的矩阵和运算都类似. 这一性能问题促使我们尝试改组楚列斯基和其他线性代数程序为更简化的版本, 如同矩阵 - 矩阵乘法一样快. 这些来自 LAPACK 改组的代码的速度在表格的最后两行中给出. 显然 LAPACK 程序非常接近机器的最大速度. 我们强调 LAPACK 和 LINPACK 楚列斯基程序以不同的次序执行相同的浮点运算.

为了理解如何达到这些加速, 必须理解当计算机执行时如何花费时间. 由此需要理解计算机存储器如何操作. 从最便宜的个人计算机到最大型的超级计算机的所有计算机存储器由具有一系列不同类型的存储器的分层结构构成, 分层结构的顶部是一些非常快速的、价格昂贵的小型存储器, 而底部则是慢速的、价格低廉的、非常大型的存储器.

64



例如, 寄存器构成最快速的存储器, 然后为高速缓冲存储器、主存储器、磁盘, 最后为最慢的、最大型的和最便宜的磁带. 在分层结构顶部寄存器中仅对数据作有用的算术运算和逻辑运算. 在分层结构存储器的一个层面的数据可移动到邻近的层面上去——例如, 移动主存储器和磁盘之间的数据. 在分层结构顶部近旁数据移动的速度是很快的(寄存器与高速缓冲存储器之间)而底部近旁是较慢的(磁盘和主存储器之间). 特别地, 所做的算术运算速度大大快于存储器分层结构内低层面之间数据转移的速度, 与层面有关, 速度超过 10 多倍或者甚至 1000 多倍. 这意味着一个设计上有毛病的算法为了执行有效的工作, 可能耗费大部分时间把数据从存储器分层结构顶部移动到寄存器, 而不是有效地做运算.

下面是一个简单算法的例子, 它花费大量时间移动数据而不是做有用的算术运算. 假如我们相加两个阶数足够大的 $n\times n$ 阶矩阵, 以致它们只能装配在存储器分层结构的一个大型慢速层面内. 为了相加, 每次必须把它们一起转移到寄存器中做加法并且把和转移回来. 因此, 执行每一个加法在快速和慢速存储器之间正好有 3 次

存储器转移(把2个加数读入快速存储器并且把1个和写回到慢速存储器). 若做一次浮点运算的时间为 t_{arith} 秒而存储器层面之间移动一个数据字的时间为 t_{mem} 秒，其中 $t_{\text{mem}} \gg t_{\text{arith}}$，则这个算法的执行时间为 $n^2(t_{\text{arith}}+3t_{\text{men}})$，这个时间比单独算术运算需要的时间 $n^2 t_{\text{arith}}$ 大得多. 这意味着矩阵加法注定以存放矩阵的存储器的最慢速层面的速度而不是快得多的加法速度运行. 相反，在后面将看到其他的运算，诸如矩阵 - 矩阵乘法，即使数据原来是被存放在最慢速的层面中，也可能以存储器的最快速层面的速度运行.

LINPACK 的楚列斯基程序运行如此之慢，是因为在诸如 Cray YMP[1] 那样的机器上它未被设计成使存储器移动达到极小. 相反，在表格中估量的矩阵 - 矩阵乘法和其他三个基本线性代数算法被限定在一台 Cary YMP 上处于数据移动极小状态.

65

2.6.1　基本线性代数子程序(BLAS)

因为为每台新的计算机编写像楚列斯基那样的每个程序的特别版本是不划算的，所以我们需要一种更有规律的方法. 因为像矩阵 - 矩阵乘法这样的运算是如此普遍，所以计算机厂商已把它们标准化为基本线性代数子程序，或称 BLAS[169,89,87]，并且针对机器做了优化. 换言之，在高性能机器上(以及许多其他机器上)可以得到有标准的 Fortran 或 C 语言界面的矩阵 - 矩阵乘法、矩阵 - 向量乘法及其他类似运算的子程序库，但是在每台机器里面它们已经被优化. 我们的目标是利用这些优化的 BLAS 的优点改组像楚列斯基那样的算法. 使得它们调用 BLAS 来执行它们的大部分工作.

在本节中将概要地讨论 BLAS. 特别地，在 2.6.2 节中描述如何优化矩阵乘法. 最后，在 2.6.3 节中指出如何改组高斯消元法，使得它的大部分工作利用矩阵乘法来执行.

让我们更仔细地考察 BLAS. 表格 2.1 计算存储器引用次数和三个相关的 BLAS 执行的浮点运算次数. 例如，在表格第一行中执行 saxpy 运算所需存储器引用次数是 $3n+1$，因为需要把慢速存储器中 n 个 x_i 值，n 个 y_i 值和一个 α 值读到寄存器，然后把 n 个 y_i 值写回到慢速存储器. 最后一列给出 flops 和存储器引用数之比 q(它的最高项仅为 n).

q 的意义是它粗略地告诉我们每个存储器访问可以执行多少 flops，或者与移动数据的时间相比可以做多少有用的工作. 这告诉我们算法潜在地可以运行多快. 例如，假如一个算法执行 f 次浮点运算，每次运算需要 t_{arith} 秒，执行 m 次存储器访问，每次访问需要 t_{mem} 秒. 假如算术运算和存储器引用不是并行地执行的，则总的运行时间为

1. 它被设计成减少另一种存储器转移，在主存储器和硬盘之间的页面断层(page faults).

$$f \cdot t_{\text{arith}} + m \cdot t_{\text{mem}} = f \cdot t_{\text{arith}} \cdot \left(1 + \frac{m\, t_{\text{mem}}}{f\, t_{\text{arith}}}\right) = f \cdot t_{\text{arith}} \cdot \left(1 + \frac{1}{q}\frac{t_{\text{mem}}}{t_{\text{arith}}}\right).$$

所以，q 的值越大，运行时间越靠近最佳运行时间 $f \cdot t_{\text{arith}}$，这个值就是当所有数据都在寄存器内时，算法所需要的时间. 这意味着具有较大 q 值的算法是其他算法较好的构造部件.

表 2-1 反映运算的分层结构：像 saxpy 这种运算对向量执行 $O(n^1)$ 次 flops 并提供最坏的 q 值，这些运算称为 1 级 BLAS 或 BLAS1[169]，它也包括内积，用一个标量乘一个向量以及其他简单的运算. 像矩阵－向量乘法那样的运算对矩阵和向量执行 $O(n^2)$ 次 flops，并提供稍好一些的 q 值，这些运算称为 2 级 BLAS 或 BLAS2 [89,88]，它也包括解三角形方程组和矩阵的秩－1 修正（$\boldsymbol{A} + \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}$，$\boldsymbol{x}$ 和 $\boldsymbol{y}$ 是列向量）. 像矩阵－矩阵乘法那样的运算对矩阵执行 $O(n^3)$ 次运算并提供最好的 q 值，这些运算称为 3 级 BLAS 或 BLAS3[87,86]，它也包括解具有多个右端项的三角形方程组. [66]

表 2-1 对 BLAS 计算浮点运算和存储器引用次数. f 是浮点运算次数，m 是存储器引用次数

运　算	定　义	f	m	$q=f/m$
saxpy (BLAS1)	$y=\alpha \cdot x+y$ 或 $y_i=\alpha x_i+y_i$ $i=1,\cdots,n$	$2n$	$3n+1$	$2/3$
矩阵－向量乘法 (BLAS2)	$y=A \cdot x+y$ 或 $y_i=\sum_{j=1}^{n} a_{ij}x_j+y_i$ $i=1,\cdots,n$	$2n^2$	n^2+3n	2
矩阵－矩阵乘法 (BLAS3)	$C=A \cdot B+C$ 或 $c_{ij}=\sum_{k=1}^{n} a_{ik}b_{kj}+c_{ij}$ $i,j=1,\cdots,n$	$2n^3$	$4n^2$	$n/2$

目录 NETLIB/blas 包括文件和 BLAS 所有的（未优化的）工具. BLAS 所有的快速摘要见 NETLIB/blas/blasqr. ps. 这个摘要也出现在[10，附录 C]（或 NETLIB/lapack/lug/lapack_lug. html）中.

因为 3 级 BLAS 有最高的 q 值，所以我们尽力用矩阵－矩阵乘法那样的运算而不是 saxpy 或矩阵－向量乘法去改组我们的算法.（LINPACK 的楚列斯基是调用 saxpy 来构造的.）我们强调只有当使用已经优化的 BLAS 时，这样改组的算法才会比较快.

2.6.2 如何优化矩阵乘法

详细考察如何实现矩阵乘法 $\boldsymbol{C}=\boldsymbol{A} \cdot \boldsymbol{B}+\boldsymbol{C}$ 使存储器传送的次数达到极小，从而优化其性能. 我们将看到性能对实现的细节是敏感的. 为简化讨论，将使用下列机

器模型. 假定矩阵像在 Fortran 语言中那样按列存放(下面一些例子中如果矩阵像 C 语言中那样按行存放，它是容易修正的). 假定存储器分层结构有快速的和慢速的两个层面，其中慢速存储器大到足够包含三个 $n\times n$ 阶矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$ 和 $\boldsymbol{C}$，但是快速存储器只包含 M 个字，其中 $2n<M\ll n^2$，这意味着快速存储器大到足够容纳矩阵的两列或两行但不能容纳整个矩阵. 进一步假定数据移动是在程序员控制之下(实际上，数据移动可能由像高速缓冲存储器控制器那样的硬件自动进行. 然而，基本的优化格式保持一致).

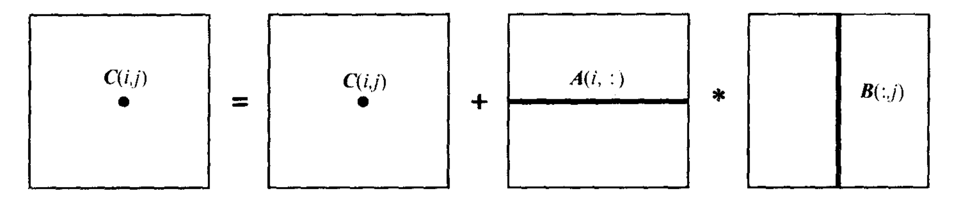
67

可尝试由三个嵌套循环组成最简单的矩阵乘法算法. 为表明数据移动，我们已对此算法作了注释.

算法 2.6 非分块的矩阵乘法(注释表明存储器所做的事情)：

```
for i = 1 to n
    {读 A 的第 i 行到快速存储器内}
    for j = 1 to n
        {读 C_ij 到快速存储器内}
        {读 B 的第 j 列到快速存储器内}
        for k = 1 to n
            C_ij = C_ij + A_ik · B_kj
        end for
        {将 C_ij 写回到慢速存储器内}
    end for
end for
```

最内层循环是为计算 $\boldsymbol{C}_{ij}$ 而做 $\boldsymbol{A}$ 的第 i 行和 $\boldsymbol{B}$ 的第 j 列的点积，如下图所示：

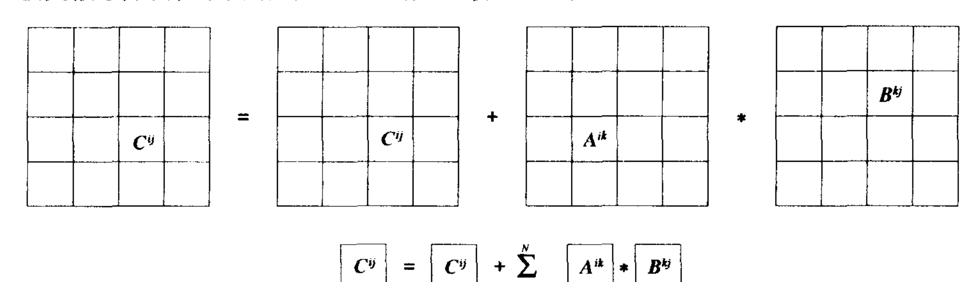


我们也可描述两个最内层的循环(关于 j 和 k)为得到 $\boldsymbol{C}$ 的第 i 行做 $\boldsymbol{A}$ 的第 i 行乘矩阵 $\boldsymbol{B}$ 的向量 - 矩阵乘法. 这暗示未执行任何优于 BLAS1 和 BLAS2 的运算，因为它们处于最内层循环之中.

下面是存储器引用的详细计算：读 $\boldsymbol{B}$ 为 n 次(每个 i 值 1 次)，共引用 n^3 次；读 $\boldsymbol{A}$ 的一列并在快速存储器中保留它直到不再需要它，共引用 n^2 次；一次读 $\boldsymbol{C}$ 的一个元并在快速存储器中保留它直到它被完全算出，然后将它传回到慢速存储器，共引用 $2n^2$ 次. 合计 n^3+3n^2 次存储器传送，或者 $q=2n^3/(n^2+3n^2)\approx 2$，它不比第 2 层次

BLAS 好，并远离最大可能值 $n/2$（见表 2-1）. 若 $M \ll n$，以致不能在快速存储器中保留 A 的整行，因为算法化为一串内积，这是第 1 层次 BLAS，所以 q 进一步减少到 1. [68] 对三个循环中的 i, j 和 k 的每一个置换，我们得到具有大约相同的 q 的另一个算法.

首选的算法利用分块，其中 $\boldsymbol{C}$ 分成具有 $n/N \times n/N$ 块 C^{ij} 的 $N \times N$ 块矩阵，$\boldsymbol{A}$ 和 $\boldsymbol{B}$ 被类似地分块，下面指出 $N=4$ 的情况. 算法变成



算法 2.7 分块矩阵乘法（注释表明存储器所做的事情）

for $i=1$ to N
  for $j=1$ to N
  {读 $\boldsymbol{C}^{ij}$ 到快速存储器内}
    for $k=1$ to N
      {读 $\boldsymbol{A}^{ik}$ 到快速存储器内}
      {读 $\boldsymbol{B}^{kj}$ 到快速存储器内}
      $\boldsymbol{C}^{ij} = \boldsymbol{C}^{ij} + \boldsymbol{A}^{ik} \cdot \boldsymbol{B}^{kj}$
    end for
  {将 $\boldsymbol{C}^{ij}$ 写回到慢速存储器内}
  end for
end for

存储器引用计算如下：读和写 $\boldsymbol{C}$ 的每个块一次共 $2n^2$ 次，读 $\boldsymbol{A}$ 为 N 次（读每个 $n/N \times n/N$ 阶子阵 $\boldsymbol{A}^{ik}$ 为 N^3 次）共 Nn^2 次，读 $\boldsymbol{B}$ 为 N 次（读每个 $n/N \times n/N$ 阶子阵 B^{kj} 为 N^3 次）共 Nn^2 次. 总共为 $(2N+2)n^2 \approx 2Nn^2$ 次存储器引用. 故为了极小化存储器引用次数希望选择 N 尽可能小. 但 N 服从约束 $M \geqslant 3(n/N)^2$，这意味着来自 $\boldsymbol{A}$、$\boldsymbol{B}$ 和 $\boldsymbol{C}$ 的每一个块必须同时装入到快速存储器中. 这就得到 $N \approx n\sqrt{3/M}$，因而 $q \approx (2n^3)/(2Nn^2) \approx \sqrt{M/3}$，这个值比前面算法的值好得多. 特别地当 M 增大时 q 增大与 n 无关，这意味着预期对任意阶数为 n 的矩阵算法是快的，并且如果快速存储器尺寸 M 增大的话，算法会变得更快. 这两者都是有吸引力的性质. [69]

事实上，可以证明算法 2.7 是渐近最优的[151]. 换言之，非重组的矩阵-矩阵

乘法(执行相同的 $2n^3$ 次算术运算)可能有大于 $O(\sqrt{M})$ 的 q.

另一方面，这个简明的分析忽略许多实际问题：

1. 一个真正的代码必须处理非方的矩阵，而其最优块尺寸可能不是方的.

2. 机器的高速缓冲存储器和寄存器结构将强有力地影响子矩阵的最佳形状.

3. 可能存在一段时间中同时执行一个乘法和一个加法的特殊的硬件指令. 如果它们互不干扰的话，它也可能同时执行几个乘-加运算.

对于一台高性能工作站 IBM RS6000/590 这些问题的详细讨论可见[1]，PARALLEL_HOMEPAGE，或 http://www.rs6000.ibm.com/resource/technology/essl.html. 图2-5 指出这台机器三种基本的 BLAS 的速度. 水平轴是矩阵尺寸，垂直轴是用 Mflops 度量的速度. 最高的机器速度是 266 Mflops. 顶部的曲线(最高的接近 250 Mflops)是方的矩阵-矩阵乘法. 中间的曲线(最高的接近 100 Mflops)是方的矩阵-向量乘法，底部的曲线(最高的接近 75 Mflops)是 saxpy. 注意对于较大的矩阵速度增大. 这是一个普遍的现象，意味着我们将尝试开发的算法其内在的矩阵乘法应使用阶数适当大的矩阵.

上面两个矩阵-矩阵乘法算法都执行 $2n^3$ 次算术运算. 其实也存在其他的使用运算次数少得多的矩阵-矩阵乘法执行过程. Strassen 方法[3]是这些算法中第一个被发现的，并且解释起来是最简单的. 这个算法将矩阵分隔为 2×2 块矩阵并用 7 次矩阵乘法(递归地)相乘子块和 18 次一半大小矩阵加法；这就导致渐近复杂性 $n^{\log_2 7}\approx n^{2.81}$，而不是 n^3.

算法 2.8　Strassen 矩阵乘法算法：

$\boldsymbol{C}=\text{Strassen}(\boldsymbol{A},\boldsymbol{B},n)$

/* 返回 $\boldsymbol{C}=\boldsymbol{A}*\boldsymbol{B}$，其中 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 是 $n\times n$ 阶矩阵；
假定 n 是 2 的幂 */

if $n=1$

返回 $\boldsymbol{C}=\boldsymbol{A}*\boldsymbol{B}$ /* 标量乘法 */

else

$$\text{分块 } \boldsymbol{A}=\begin{bmatrix}\boldsymbol{A}_{11} & \boldsymbol{A}_{12}\\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22}\end{bmatrix} \text{和 } \boldsymbol{B}=\begin{bmatrix}\boldsymbol{B}_{11} & \boldsymbol{B}_{12}\\ \boldsymbol{B}_{21} & \boldsymbol{B}_{22}\end{bmatrix}$$

其中子块 $\boldsymbol{A}_{ij}$ 和 $\boldsymbol{B}_{ij}$ 是 $n/2\times n/2$ 阶矩阵

$\boldsymbol{P}_1=\text{Strassen}(\boldsymbol{A}_{12}-\boldsymbol{A}_{22},\boldsymbol{B}_{21}+\boldsymbol{B}_{22},n/2)$

$\boldsymbol{P}_2=\text{Strassen}(\boldsymbol{A}_{11}+\boldsymbol{A}_{22},\boldsymbol{B}_{11}+\boldsymbol{B}_{22},n/2)$

$\boldsymbol{P}_3=\text{Strassen}(\boldsymbol{A}_{11}-\boldsymbol{A}_{21},\boldsymbol{B}_{11}+\boldsymbol{B}_{12},n/2)$

$\boldsymbol{P}_4=\text{Strassen}(\boldsymbol{A}_{11}+\boldsymbol{A}_{12},\boldsymbol{B}_{22},n/2)$

70

（续）

$P_5 = \text{Strassen}(A_{11}, B_{12} - B_{22}, n/2)$

$P_6 = \text{Strassen}(A_{22}, B_{21} - B_{11}, n/2)$

$P_7 = \text{Strassen}(A_{21} + A_{22}, B_{11}, n/2)$

$C_{11} = P_1 + P_2 - P_4 + P_6$

$C_{12} = P_4 + P_5$

$C_{21} = P_6 + P_7$

$C_{22} = P_2 - P_3 + P_5 - P_7$

返回 $C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$

end if

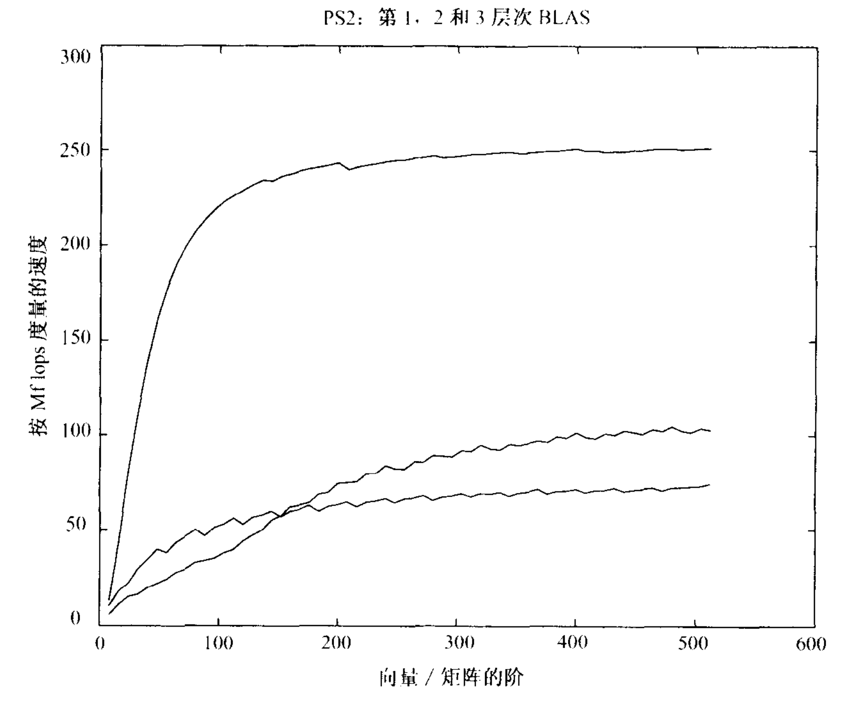


图2-5　在IBM RS6000/590上的BLAS速度

用归纳法确认这个算法可正确地做矩阵乘法是乏味的，但是直接了当（见问题2.21）．为证明它的复杂性是 $O(n^{\log_2 7})$，设 $T(n)$ 是算法执行的加法、减法和乘法的次数．因为算法执行7次递归调用 $n/2$ 阶矩阵，和18次 $n/2 \times n/2$ 阶矩阵的加法，所以

[71] 可以把递归写成 $T(n)=7T(n/2)+18(n/2)^2$. 把变量 n 改变为 $m=\log_2 n$, 得到一个新的递归 $\overline{T}(m)=7\overline{T}(m-1)+18(2^{m-1})^2$, 其中 $\overline{T}(m)=T(2^m)$. 可以进一步证实这个 $\overline{T}$ 的线性递归有一个解 $\overline{T}(m)=O(7^m)=O(n^{\log_2 7})$.

Strassen 算法的价值不仅仅是这个渐近复杂性，而且它把问题简化为一个更小的最终能装入快速存储器的子问题. 一旦子问题装入快速存储器，就可使用标准的矩阵乘法. 这个方法已导致在某种机器上对相对大的矩阵运算的加速[22]. 虽然它可用于许多用途[77]，但是它的缺点是需要相当数量的工作空间以及它具有相当低的数值稳定性. 有许多其他的甚至更快的矩阵乘法，目前的记录是由 Winograd 和 Coppersmith[263]所给出的大约为 $O(n^{2.376})$. 但是对不切实际地大的 n 值这些算法仅仅比 Strassen 算法执行较少的运算. 综述可见[195].

2.6.3　使用3级 BLAS 改组高斯消元法

首先使用2级 BLAS，然后再使用3级 BLAS 改组高斯消元法. 为简单起见，假定不选主元是必要的.

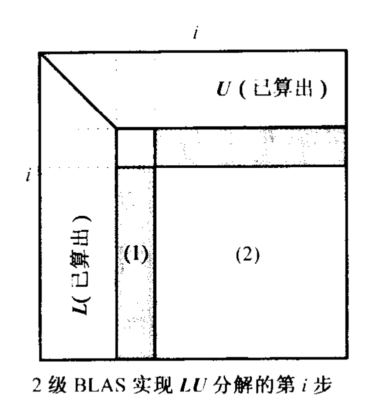
事实上，算法 2.4 已经是一个 2 级 BLAS 算法，因为大部分工作是在第 2 行做的，$A(i+1:n,i+1:n)=A(i+1:n,i+1:n)-A(i+1:n,i)*A(i,i+1:n)$,这是子矩阵 $A(i+1:n,i+1:n)$的一个秩 -1 修正. 算法中其他的算术运算 $A(i+1:n,i)=A(i+1:n,i)/A(i,i)$ 实际上是用标量 $1/A(i,i)$乘以向量 $A(i+1:n,i)$来完成的,因为乘法比除法快得多. 这也是一个 1 级 BLAS 运算,因为我们将在 3 级版本中使用它,所以必须稍微修改一下算法 2.4.

算法 2.9　对 $m\times n$ 阶矩阵 A 不选主元 LU 分解的 2 级 BLAS 执行过程，其中 $m\geq n$. 用$m\times n$阶矩阵 L 和 $m\times m$ 阶矩阵 U 覆盖 A. 我们对重要的行编了号，便于后面提及.

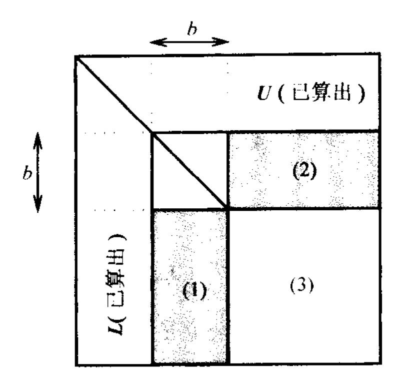
```
for i = 1 to min(m-1,n)
(1)      A(i+1:m,i) = A(i+1:m,i)/A(i,i)
         if i < n
(2)        A(i+1:m,i+1:n) = A(i+1:m,i+1:n) -
                 A(i+1:m,i) · A(i,i+1:n)
end for
```

图2-6 的左边说明算法 2.9 应用于一个方阵. 在算法的第 i 步，L 的第 1 列到第 $i-1$列和 U 的第 1 行到第 $i-1$ 行已算出. 要计算的是 L 的第 i 列和 U 的第 i 行，A 的尾部的子阵用一个秩-1 修正来更新. 图 2-6 的右边，用更新它们算法的行[(1)或(2)]来标记子矩阵. 在行(2)中的秩-1 修正是减去来自标记(2)的子矩阵中带阴影的列和带阴影的行之积.

[72]



2 级 BLAS 实现 LU 分解的第 i 步



3 级 BLAS 实现 LU 分解的第 i 步

图 2-6　2 级和 3 级 BLAS 实现 LU 分解

3 级 BLAS 算法将通过对子矩阵（2）b 步的延迟和更新来改组这个计算，其中 b 是一个小的整数，称为块大小(block size)，稍后在一次单独的乘法中应用 b 个秩-1 立刻更新全部．为看看如何做这件事，假如已经计算 $\boldsymbol{L}$ 的前 $i-1$ 列和 $\boldsymbol{U}$ 的前 $i-1$ 行，得到

$$\boldsymbol{A}=\begin{matrix} i-1 \\ b \\ n-b-i+1 \end{matrix}\overset{\begin{matrix} i-1 & b & n-b-i+1 \end{matrix}}{\begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \boldsymbol{A}_{13} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \boldsymbol{A}_{23} \\ \boldsymbol{A}_{31} & \boldsymbol{A}_{32} & \boldsymbol{A}_{33} \end{pmatrix}}$$

$$=\begin{bmatrix} \boldsymbol{L}_{11} & 0 & 0 \\ \boldsymbol{L}_{21} & \boldsymbol{I} & 0 \\ \boldsymbol{L}_{31} & 0 & \boldsymbol{I} \end{bmatrix}\cdot\begin{bmatrix} \boldsymbol{U}_{11} & \boldsymbol{U}_{21} & \boldsymbol{U}_{31} \\ 0 & \widetilde{\boldsymbol{A}}_{22} & \widetilde{\boldsymbol{A}}_{23} \\ 0 & \widetilde{\boldsymbol{A}}_{32} & \widetilde{\boldsymbol{A}}_{33} \end{bmatrix}$$

其中所有的矩阵以相同的方式分块，这被表示在图 2-6 的右边．现在应用算法 2.9 于子矩阵$\begin{bmatrix} \widetilde{\boldsymbol{A}}_{22} \\ \widetilde{\boldsymbol{A}}_{32} \end{bmatrix}$，得到

$$\begin{bmatrix} \widetilde{\boldsymbol{A}}_{22} \\ \widetilde{\boldsymbol{A}}_{32} \end{bmatrix}=\begin{bmatrix} \boldsymbol{L}_{22} \\ \boldsymbol{L}_{32} \end{bmatrix}\cdot\boldsymbol{U}_{22}=\begin{bmatrix} \boldsymbol{L}_{22} & \boldsymbol{U}_{22} \\ \boldsymbol{L}_{32} & \boldsymbol{U}_{22} \end{bmatrix}$$

这可写成

73

$$\begin{bmatrix}\widetilde{A}_{22} & \widetilde{A}_{23}\\ \widetilde{A}_{32} & \widetilde{A}_{33}\end{bmatrix}=\begin{bmatrix}L_{22}U_{22} & \widetilde{A}_{23}\\ L_{32}U_{22} & \widetilde{A}_{33}\end{bmatrix}=\begin{bmatrix}L_{22} & 0\\ L_{32} & I\end{bmatrix}\cdot\begin{bmatrix}U_{22} & L_{22}^{-1}\widetilde{A}_{23}\\ 0 & \widetilde{A}_{33}-L_{32}\cdot(L_{22}^{-1}\widetilde{A}_{23})\end{bmatrix}$$

$$\equiv\begin{bmatrix}L_{22} & 0\\ L_{32} & I\end{bmatrix}\cdot\begin{bmatrix}U_{22} & U_{23}\\ 0 & \widetilde{A}_{33}-L_{32}\cdot U_{23}\end{bmatrix}\equiv\begin{bmatrix}L_{22} & 0\\ L_{32} & I\end{bmatrix}\cdot\begin{bmatrix}U_{22} & U_{23}\\ 0 & \widetilde{\widetilde{A}}_{33}\end{bmatrix}.$$

合在一起，得到一个完整的关于 L 另外的 b 列和 U 另外的 b 行更新的分解：

$$\begin{bmatrix}A_{11} & A_{12} & A_{13}\\ A_{21} & A_{22} & A_{23}\\ A_{31} & A_{32} & A_{33}\end{bmatrix}=\begin{bmatrix}L_{11} & 0 & 0\\ L_{21} & L_{22} & 0\\ L_{31} & L_{23} & I\end{bmatrix}\cdot\begin{bmatrix}U_{11} & U_{21} & U_{31}\\ 0 & U_{22} & U_{23}\\ 0 & 0 & \widetilde{\widetilde{A}}_{33}\end{bmatrix}.$$

这就定义了一个具有下列三步的算法，在图2-6的右边说明了：

(1) 利用算法2.9分解 $\begin{bmatrix}\widetilde{A}_{22}\\ \widetilde{A}_{32}\end{bmatrix}=\begin{bmatrix}L_{22}\\ L_{32}\end{bmatrix}\cdot U_{22}$.

(2) 构成 $U_{23}=L_{22}^{-1}\widetilde{A}_{23}$. 这意味着以一个单独的3级BLAS运算解一个具有多个右端项($\widetilde{A}_{23}$)的三角形线性方程组.

(3) 一个矩阵-矩阵乘法，构成 $\widetilde{\widetilde{A}}_{33}=\widetilde{A}_{33}-L_{32}\cdot U_{23}$.

更正式地有下列算法.

> **算法2.10** 对 $n\times n$ 阶矩阵 A 不选主元 LU 分解的3级BLAS执行过程. 算法的行如上面那样标号，并对应于图2-6的右边.
>
> for $i=1$ to $n-1$ step b
>
> (1) 利用算法2.9分解 $A(i:n,i:i+b-1)=\begin{bmatrix}L_{22}\\ L_{32}\end{bmatrix}U_{22}$
>
> (2) $A(i:i+b-1,i+b:n)=L_{22}^{-1}\cdot A(i:i+b-1,i+b:n)$
>
> /* 构成 U_{23} */
>
> (3) $A(i+b:n,i+b:n)=A(i+b:n,i+b:n)$
> $-A(i+b:n,i:i+b-1)\cdot A(i:i+b-1,i+b:n)$
>
> /* 构成 $\widetilde{\widetilde{A}}_{33}$ */
>
> end for

为了极大化算法的速度，我们仍旧需要选择块大小 b. 一方面，因为已经看到当较大的矩阵相乘时速度增快，所以喜欢使 b 大些. 另一方面，可以验证在算法的行(1) 中用较慢的 2 级和 1 级 BLAS 执行的浮点运算次数，对小的 b 大约为 $n^2b/2$，当 b 增长时它增长，故我们不希望选择 b 太大. b 的最优值与机器相关，并且对每台机器可以调整. 一般使用 $b=32$ 或 $b=64$.

74

为仔细地观察算法 2.9 和 2.10 的执行过程，分别可见 LAPACK(NETLIB/lapack)中的 sgetf2 和 sgetrf. 关于块算法，包括在多种机器上详尽的执行次数更多的信息也可见[10]或在 PARALLEL_HOMEPAGE 上的课程注释.

2.6.4 更多的并行性和其他性能问题

本节简要地综述尽可能有效地执行高斯消元法(及其他线性代数程序)中所涉及的其他问题.

并行计算机包含 $p>1$ 台处理器，可同时处理同一个问题. 人们可能希望在这样一台机器上求解问题比在一台传统的单处理器机器上快 p 倍. 但这种"完美的效率"很少能达到，即使始终至少有 p 个独立的任务可做，因为还有协调 p 台处理器的管理开销，以及把数据从存放的处理器中送到需要的处理器中的代价. 这个最后的问题是存储器分层结构的另一个例子：从处理器 i 的角度看，它自己的存储器是最快的，但通过处理机 j 的存储器得到数据是较慢的，有时慢几千倍.

高斯消元法提供许多并行机会，因为在每一步尾部子阵的元素可以独立地和并行地更新. 然而，为了尽可能地有效，还是需要注意一些细节. 两个标准的软件块是可利用的，倘若有并行运行的 BLAS 的有效执行过程，上节中描述的 LAPACK 程序 sgetrf[10]在共享存储器并行机上运行. 对可升级的 LAPACK [34,53]，一个称为 ScaLAPACK 的有关的库，是为分布式存储器并行机设计的. 那些机器在不同的处理机之间移动数据需要特殊的运算. 所有的软件在 LAPACK 和 ScaLAPACK 子目录中的 NETLIB 上是可以得到的. ScaLAPACK 在 PARALLEL_HOMEPAGE 的注释中被更详尽地描述. 线性方程解法器的扩展性能数据像 LINPACK Benchmark[85]一样可以利用，在 NETLIB/benchmark/performance.ps 上或者在性能数据库服务器[1](Performance Database Server)上可以找到一个最新的版本. 到 1997 年 5 月为止，用高斯消元法求解任意线性方程组最快的是在一台有 $p=7264$ 个处理机的 Intel ASCI Option Red 上对 $n=215\ 000$ 的一个方程组. 问题运行刚超过 1068Gflops(千兆 flops)，在最大的 1453Gflops 之外.

75

有些矩阵太大以致于不适合装入到任何可利用的机器的主存储器中. 这些矩阵被存放在硬盘中，为了执行高斯消元法必须逐步地读入到主存储器中. 这种程序的很大程度上用上面描述的技巧组织起来，它们被包含在 ScaLAPACK 中.

1. http://performance.netlib.org/performance/html/PDStop.html.

最后，人们可能希望编译器变得足够聪明，采取利用三个嵌套循环的高斯消元法的最简单的执行过程，并且看来像上面小节中讨论的类似的块算法自动地“优化”代码. 虽然对这个论题有许多最新的研究成果(见最新的编译器教科书[264]中的参考文献)，但是还没有能与 LAPACK 和 ScaLAPACK 那样优化的库相提并论的快速而可靠的软件.

2.7 特殊的线性方程组

正如1.2中提到的，为了增加解的速度和减少存储量，重要的是充分利用矩阵所有的特殊结构. 当然实际上必须考虑为了充分利用这个结构所需要的额外的程序设计工作量. 例如，如果目标就是尽快得到解，为了把求解时间从10秒减少到1秒，它需要额外的一周程序设计工作，所以只有当我们打算使用这个程序多于(1 星期 * 7 天/星期 * 24 小时/天 * 3 600 秒/小时)/(10 秒 - 1 秒) = 67 200 次时，这项工作才是值得做的. 所幸，有一些频繁出现的特殊结构表明存在标准解，我们确实应该利用它们. 这里我们考虑的是

1. s. p. d 矩阵;
2. 对称不定矩阵;
3. 带状矩阵;
4. 一般稀疏矩阵;
5. 由少于 n^2 个独立参数决定的稠密矩阵.

我们将只考虑实矩阵，推广到复矩阵是直接了当的.

2.7.1 实对称正定矩阵

记得实矩阵 $\boldsymbol{A}$ 是 s. p. d. 当且仅当 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$ 和对一切 $x\neq 0$，$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}>0$. 本节中将证明当 $\boldsymbol{A}$ 是 s. p. d. 时如何以高斯消元法的一半时间和一半空间求解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$.

76

命题2.2 1. *若 $\boldsymbol{X}$ 非奇异，则 $\boldsymbol{A}$ 是 s. p. d. 当且仅当 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}$ 是 s. p. d..*

2. *若 $\boldsymbol{A}$ 是 s. p. d. 且 $\boldsymbol{H}$ 是 $\boldsymbol{A}$ 的主子矩阵[$\boldsymbol{H}=\boldsymbol{A}(j:k,j:k)$,对某些 $j\leqslant k$]，则 $\boldsymbol{H}$ 是 s. p. d..*

3. *$\boldsymbol{A}$ 是 s. p. d. 当且仅当 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$ 且它的所有特征值是正的.*

4. *若 $\boldsymbol{A}$ 是 s. p. d.，则所有的 $a_{ii}>0$ 且 $\max_{ij}|a_{ij}|=\max_i a_{ii}>0$.*

5. *$\boldsymbol{A}$ 是 s. p. d. 当且仅当存在唯一的一个具有正对角元的下三角形非奇异阵 $\boldsymbol{L}$，使得 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$. $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 称为 $\boldsymbol{A}$ 的楚列斯基分解，$\boldsymbol{L}$ 称为 $\boldsymbol{A}$ 的楚列斯基因子.*

证明

(1) $\boldsymbol{X}$ 非奇异推出对所有 $\boldsymbol{x}\neq 0,\boldsymbol{X}\boldsymbol{x}\neq 0$，故对所有 $\boldsymbol{x}\neq 0,\boldsymbol{x}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}\boldsymbol{x}>0$. 因而 $\boldsymbol{A}$ 是 s. p. d. 推出 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}$ 是 s. p. d. 利用 $\boldsymbol{X}^{-1}$ 推出其余的结论.

(2) 首先假定 $\boldsymbol{H}=\boldsymbol{A}(1:m,1:m)$. 则给定任意的 m 维向量 $\boldsymbol{y}$，n 维向量 $\boldsymbol{x}=$

$[y^{\mathrm{T}},0]^{\mathrm{T}}$ 满足 $y^{\mathrm{T}}Hy = x^{\mathrm{T}}Ax$. 因而若对所有非零的 x, $x^{\mathrm{T}}Ax>0$, 则对所有非零的 y, $y^{\mathrm{T}}Hy>0$,故 H 是 s. p. d. 若 H 不位于 A 的左上角，设 P 是一个置换阵使得 H 位于 $P^{\mathrm{T}}AP$ 的左上角，然后应用第 1 部分来证明.

(3)设 X 是 A 的实正交特征向量矩阵使得 $X^{\mathrm{T}}AX=\Lambda$ 是实特征值 λ_i 的对角阵. 因为 $x^{\mathrm{T}}\Lambda x = \sum_i \lambda_i x_i^2$, 所以 Λ 是 s. p. d. 当且仅当每个 $\lambda_i>0$. 然后应用第 1 部分来证明.

(4)设 e_i 是单位矩阵的第 i 列. 则对所有的 i, $e_i^{\mathrm{T}}Ae_i=a_{ii}>0$. 若 $|a_{kl}|=\max_{ij}|a_{ij}|$, 但 $k\neq l$, 选择 $x=e_k-\mathrm{sign}(a_{kl})e_l$. 则 $x^{\mathrm{T}}Ax=a_{kk}+a_{ll}-2|a_{kl}|\leqslant 0$, 与正定性矛盾.

(5)假定 $A=LL^{\mathrm{T}}$, L 非奇异. 则对所有的 $x\neq 0$, $x^{\mathrm{T}}Ax=(x^{\mathrm{T}}L)(L^{\mathrm{T}}x)=\|L^{\mathrm{T}}x\|_2^2>0$,故 A 是 s. p. d.. 若 A 是 s. p. d.，我们对维数 n 用归纳法证明 L 存在. 若选择每个 $l_{ii}>0$，则我们的构造将唯一地确定 L. 若 $n=1$，选择 $l_{11}=\sqrt{a_{11}}$，因为 $a_{11}>0$，所以这是存在的. 与高斯消元法一样，了解 2×2 块情况就足够了. 记

$$
\begin{aligned}
A &= \begin{bmatrix} a_{11} & A_{12} \\ A_{12}^{\mathrm{T}} & A_{22} \end{bmatrix} \\
&= \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \dfrac{A_{12}^{\mathrm{T}}}{\sqrt{a_{11}}} & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{A}_{22} \end{bmatrix} \begin{bmatrix} \sqrt{a_{11}} & \dfrac{A_{12}}{\sqrt{a_{11}}} \\ 0 & I \end{bmatrix} \\
&= \begin{bmatrix} a_{11} & A_{12} \\ A_{12}^{\mathrm{T}} & \widetilde{A}_{22} + \dfrac{A_{12}^{\mathrm{T}}A_{12}}{a_{11}} \end{bmatrix}
\end{aligned}
$$

故 $(n-1)\times(n-1)$ 矩阵 $\widetilde{A}_{22}=A_{22}-\dfrac{A_{12}^{\mathrm{T}}A_{12}}{a_{11}}$ 对称.

由上面的第 1 部分, $\begin{bmatrix} 1 & 0 \\ 0 & \widetilde{A}_{22} \end{bmatrix}$ 是 s. p. d. 的，故由第 2 部分 $\widetilde{A}_{22}$ 也是 s. p. d. 的.

于是由归纳法知，存在一个 $\widetilde{L}$ 使得 $\widetilde{A}_{22}=\widetilde{L}\widetilde{L}^{\mathrm{T}}$ 且

$$
\begin{aligned}
A &= \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \dfrac{A_{12}^{\mathrm{T}}}{\sqrt{a_{11}}} & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{L}\widetilde{L}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \sqrt{a_{11}} & \dfrac{\sqrt{A_{12}}}{\sqrt{a_{11}}} \\ 0 & I \end{bmatrix} \\
&= \begin{bmatrix} \sqrt{a_{11}} & 0 \\ \dfrac{A_{12}^{\mathrm{T}}}{\sqrt{a_{11}}} & \widetilde{L} \end{bmatrix} \begin{bmatrix} \sqrt{a_{11}} & \dfrac{A_{12}}{\sqrt{a_{11}}} \\ 0 & \widetilde{L}^{\mathrm{T}} \end{bmatrix} \equiv LL^{\mathrm{T}}.
\end{aligned}
$$

□

77

可以改写这个归纳法为下列算法：

算法2.11　楚列斯基算法：

for $j=1$ to n

　$l_{jj}=(a_{jj}-\sum_{k=1}^{j-1}l_{jk}^2)^{1/2}$

　for $i=j+1$ to n

　　$l_{ij}=(a_{ij}-\sum_{k=1}^{j-1}l_{ik}l_{jk})/l_{jj}$

　end for

end for

若$\boldsymbol{A}$不是正定的，则（用精确的算术运算）试图计算一个负数的平方根或用零作除数，这个算法将失败；如果一个对称阵是正定的，则这是一个最廉价的方法.

与高斯消元法一样，$\boldsymbol{L}$可以覆盖在$\boldsymbol{A}$的下半部分. 由算法知只涉及$\boldsymbol{A}$的下半部分，所以事实上只需要$n(n+1)/2$而不是n^2个存储量. flops数目为

$$\sum_{j=1}^{n}(2j+\sum_{i=j+1}^{n}2j)=\frac{1}{3}n^3+O(n^2),$$

或者说刚好为高斯消元法的flops的一半. 正如高斯消元法一样，可以利用3级BLAS改组楚列斯基算法去执行大部分浮点运算. 见LAPACK程序 spotrf.

因为楚列斯基算法是数值稳定的，所以不必选主元（等价地，也可以说任何对角线选主元次序都是数值稳定的）. 这可证明如下. 如2.4.2节中对高斯消元法同样的分析指出计算解$\hat{\boldsymbol{x}}$满足$(\boldsymbol{A}+\delta\boldsymbol{A})\hat{\boldsymbol{x}}=\boldsymbol{b}$，$|\delta\boldsymbol{A}|\leqslant 3n\varepsilon|\boldsymbol{L}|\cdot|\boldsymbol{L}^{\mathrm{T}}|$. 仅仅利用柯西－施瓦茨不等式和命题2.2的第4部分

[78]

$$\begin{aligned}(|\boldsymbol{L}|\cdot|\boldsymbol{L}^{\mathrm{T}}|)_{ij}&=\sum_k|l_{ik}|\cdot|l_{jk}|\leqslant\sqrt{\sum l_{ik}^2}\sqrt{\sum l_{jk}^2}\\&=\sqrt{a_{ii}}\cdot\sqrt{a_{jj}}\leqslant\max_{ij}|a_{ij}|,\end{aligned}\tag{2.16}$$

故$\|\,|\boldsymbol{L}|\cdot|\boldsymbol{L}^{\mathrm{T}}|\,\|_\infty\leqslant n\|\boldsymbol{A}\|_\infty$，且$\|\delta\boldsymbol{A}\|_\infty\leqslant 3n^2\varepsilon\|\boldsymbol{A}\|_\infty$.

2.7.2　对称不定矩阵

当求解一个对称而不定（既非正定又非负定）的线性方程组时自然产生我们能否仍旧节省一半时间和一半空间的问题. 结果是可能的，但需要更复杂的选主元格式和分解. 若$\boldsymbol{A}$非奇异，则可以证明存在一个置换阵$\boldsymbol{P}$，一个单位下三角形阵$\boldsymbol{L}$和一个具有1×1和2×2块的对角阵$\boldsymbol{D}$使得$\boldsymbol{PAP}^{\mathrm{T}}=\boldsymbol{LDL}^{\mathrm{T}}$. 为了观察为什么$\boldsymbol{D}$中需要$2\times2$块，考虑矩阵$\begin{bmatrix}0&1\\1&0\end{bmatrix}$. 可以稳定地算出这个分解，与标准的高斯消元法比较节省大约一半工作和空间. 做这个运算的LAPACK子程序的名称是 ssysv. 在[44]中描述

了这个算法.

2.7.3 带状矩阵

矩阵 $\boldsymbol{A}$ 称为有下带宽 b_L 和上带宽 b_U 的带状阵，如果当 $i>j+b_L$ 或 $i<j-b_U$ 时 $a_{ij}=0$：

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & \cdots & a_{1,b_U+1} & & & 0 \\ \vdots & & & a_{2,b_U+2} & & \\ a_{b_L+1,1} & & & & \ddots & \\ & a_{b_L+2,2} & & & & a_{n-b_U,n} \\ & & \ddots & & & \vdots \\ 0 & & & a_{n,n-b_L} & \cdots & a_{n,n} \end{bmatrix}$$

带状矩阵经常在实践中产生并公认是有用的，因为它们的 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 因子也是“基本上带状”的，这使得它们得以廉价地计算和存储. 下面说明“基本上带状”的含义. 但是，首先考虑不选主元的 $\boldsymbol{LU}$ 分解并指出在通常的意义下 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 是具有像 $\boldsymbol{A}$ 那样带宽的带状矩阵.

79

命题 2.3 设 $\boldsymbol{A}$ 是具有下带宽 b_L 和上带宽 b_U 的带状矩阵. 设 $\boldsymbol{A}=\boldsymbol{LU}$ 是不选主元得到的. 则 $\boldsymbol{L}$ 有下带宽 b_L 且 $\boldsymbol{U}$ 有上带宽 b_U. 当 b_U 和 b_L 小于 n 的时候，$\boldsymbol{L}$ 和 $\boldsymbol{U}$ 可用大约 $2n\cdot b_U\cdot b_L$ 次算术运算算出. 需要的空间是 (b_L+b_U+1). 求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 总共花费是 $2nb_U\cdot b_L+2nb_U+2nb_L$.

证明的要点 看一步就足够了，见图 2-7. 在高斯消元法的第 j 步，通过减去带阴影区域的第一列和第一行之积修正带阴影的区域. 注意这样做不扩大带宽. □

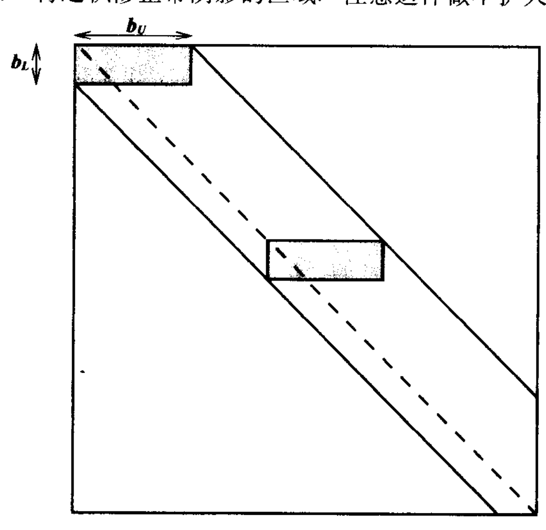


图 2-7 不选主元的带状 $\boldsymbol{LU}$ 分解

命题2.4 设A是下带宽为b_L和上带宽为b_U的带状阵. 则用选主元高斯消元法后, U是上带宽最多为b_L+b_U的带状阵, 而L是具有下带宽b_L的“基本上带状阵”. 这意味着L在每列中最多有b_L+1个非零元. 故L也能放在像具有下带宽b_L带状阵那样相同的空间中.

证明的要点 再次用算法的一步区域改变的图画来说明证明. 如图2-8中所说明的, 选主元最多能增加上带宽b_L. 后面的置换可以重新安排早先的列的元素, 使得L的元素位于次对角线b_L之下, 而不引进新的非零元, 故L所需的存储量保持每列为b_L. □

80

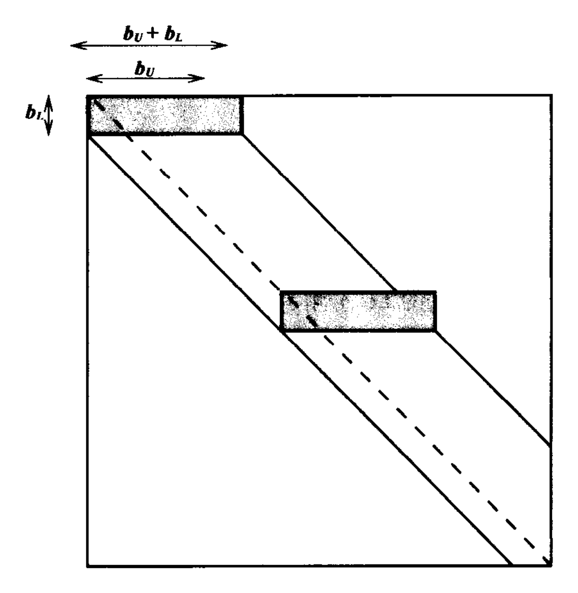


图2-8 选主元的带状 *LU* 分解

带状阵的高斯消元法和楚列斯基分解在像 ssbsv 和 sspsv 那样的 LAPACK 程序中可以得到.

带状阵通常从离散化物理问题中产生, 这些问题在网格上位置最邻近的点有交互作用(倘若未知量按行或按列排序; 也可见例2.9和6.3节).

例2.8 考虑常微分方程(ODE) $y''(x)-p(x)y'(x)-q(x)y(x)=r(x)$在区间$[a,b]$上具有边界条件$y(a)=\alpha, y(b)=\beta$. 也假设$q(x)\geq \underline{q}>0$. 例如, 这个方程可用作为细长杆的热流模型. 为数值求解微分方程, 我们离散它, 仅在等距网格点$x_i=a+ih, i=0,\cdots,N+1$上求它的解. 其中$h=(b-a)/(N+1)$是网格间距. 定义$p_i=p(x_i), r_i=r(x_i), q_i=q(x_i)$. 为了求解所要求的近似$y_i\approx y(x_i)$, 其中$y_0=\alpha$和$y_{N+1}=\beta$, 我们必须导出方程. 为导出这些方程, 用下列有限差分近似来逼近导数$y'(x_i)$:

$$y'(x_i)\approx\frac{y_{i+1}-y_{i-1}}{2h}.$$

[注意当h变得较小时, 右端的式子越来越精确地逼近$y'(x_i)$.]类似地可用下式逼近

二阶导数

$$y''(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

更详细的推导可见第6章6.3.1节.

把这些近似代入到微分方程中得到

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - p_i \frac{y_{i+1} - y_{i-1}}{2h} - q_i y_i = r_i, 1 \leqslant i \leqslant N.$$

重写此式为一个线性方程组，得到 $\boldsymbol{Ay} = \boldsymbol{b}$，其中

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \\ \vdots \\ \\ y_N \end{bmatrix}, \quad \boldsymbol{b} = \frac{-h^2}{2} \begin{bmatrix} r_1 \\ \\ \vdots \\ \\ r_N \end{bmatrix} + \begin{bmatrix} (\frac{1}{2} + \frac{h}{4} p_1)\alpha \\ 0 \\ \vdots \\ 0 \\ (\frac{1}{2} - \frac{h}{4} p_N)\beta \end{bmatrix},$$

以及

$$\boldsymbol{A} = \begin{bmatrix} a_1 & -c_1 & & \\ -b_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_{N-1} \\ & & -b_N & a_N \end{bmatrix}, \quad \begin{aligned} a_i &= 1 + \frac{h^2}{2} q_i, \\ b_i &= \frac{1}{2}\left[1 + \frac{h}{2} p_i\right], \\ c_i &= \frac{1}{2}\left[1 - \frac{h}{2} p_i\right], \end{aligned}$$

注意 $a_i > 0$ 且当 h 足够小时，$b_i > 0$ 且 $c_i > 0$.

这是一个求解 $\boldsymbol{y}$ 的非对称三对角方程组. 我们将指出如何把它变为一个对称正定三对角方程组，从而可以利用带状楚列斯基(band Cholesky)求解它.

选择 $\boldsymbol{D} = \mathrm{diag}\left(1, \sqrt{\frac{c_1}{b_2}}, \sqrt{\frac{c_1 c_2}{b_2 b_3}}, \cdots, \sqrt{\frac{c_1 c_2 \cdots c_{N-1}}{b_2 b_3 \cdots b_N}}\right)$. 于是可把 $\boldsymbol{Ay} = \boldsymbol{b}$ 变为 $(\boldsymbol{DAD}^{-1})(\boldsymbol{Dy}) = \boldsymbol{Db}$ 或 $\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{y}} = \widetilde{\boldsymbol{b}}$，其中

$$\widetilde{\boldsymbol{A}} = \begin{bmatrix} a_1 & -\sqrt{c_1 b_2} & & & \\ -\sqrt{c_1 b_2} & a_2 & -\sqrt{c_2 b_3} & & \\ & -\sqrt{c_2 b_3} & \ddots & & \\ & \ddots & \ddots & -\sqrt{c_{N-1} b_N} \\ & & -\sqrt{c_{N-1} b_N} & a_N \end{bmatrix}$$

容易看出 $\widetilde{\boldsymbol{A}}$ 是对称的，并且因为 $\boldsymbol{A}$ 和 $\widetilde{\boldsymbol{A}} = \boldsymbol{DAD}^{-1}$ 相似，$\widetilde{\boldsymbol{A}}$ 与 $\boldsymbol{A}$ 有相同的特

征值.（细节见第4.2节.）我们将利用下面的定理证明它也是正定的.

定理2.9 Gershgorin.　设 $\boldsymbol{B}$ 是一个任意的矩阵. 则 $\boldsymbol{B}$ 的特征值 λ 位于 n 个圆盘

[82]

$$|\lambda - b_{kk}| \leqslant \sum_{j \neq k} |b_{kj}|$$

的并集中.

证明　给定 λ 和 $\boldsymbol{x} \neq 0$ 使得 $\boldsymbol{Bx} = \lambda \boldsymbol{x}$，设 $1 = \|\boldsymbol{x}\|_\infty = x_k$，如果必要的话可按比例缩小 $\boldsymbol{x}$. 于是 $\sum_{j=1}^{N} b_{kj}x_j = \lambda x_k = \lambda$，故 $\lambda - b_{kk} = \sum_{\substack{j=1 \\ j \neq k}}^{N} b_{kj}x_j$，推得

$$|\lambda - b_{kk}| \leqslant \sum_{j \neq k} |b_{kj}x_j| \leqslant \sum_{j \neq k} |b_{kj}|. \qquad \square$$

若 h 对所有的 i 是如此之小，有 $\left|\frac{h}{2}p_i\right| < 1$，则

$$|b_i| + |c_i| = \frac{1}{2}\left(1 + \frac{h}{2}p_i\right) + \frac{1}{2}\left(1 - \frac{h}{2}p_i\right) = 1 < 1 + \frac{h^2}{2}\underline{q} \leqslant 1 + \frac{h^2}{2}q_i = a_i.$$

所以 $\boldsymbol{A}$ 的所有特征值位于中心在 $1 + h^2 q_i/2 \geqslant 1 + h^2\underline{q}/2$、半径为1的圆盘中；特别地，它们肯定都有正实部. 因为 $\widetilde{\boldsymbol{A}}$ 是对称的，它的特征值是实数，因此是正的，所以 $\widetilde{\boldsymbol{A}}$ 是正定的. 它的最小特征值的下界是 $\underline{q}h^2/2$. 因而，它可用楚列斯基分解求解. 求解对称正定三对角方程组的LAPACK子程序是 sptsv.　◇

在4.3节中将再次利用Gershgorin定理计算矩阵特征值的扰动界.

2.7.4　一般的稀疏阵

稀疏阵被定义为一个具有大量零元的矩阵. 实际上，这意味着一个具有很多零元的矩阵值得利用一个避免存放和运算零元的算法. 第6章专门讨论除了高斯消元法及其变形之外求解稀疏线性方程组的方法. 有大量的稀疏方法，而选择最佳的方法通常需要丰富的矩阵知识[24]. 本节中仅略述稀疏高斯消去法的基本结果，并给出参考文献和可用软件的建议.

为给出一个非常简单的例子，考虑下列矩阵，它已排序好，GEPP不用作任何行置换：

$$\boldsymbol{A} = \begin{bmatrix} 1 & & & & 0.1 \\ & 1 & & & 0.1 \\ & & 1 & & 0.1 \\ & & & 1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 1 \end{bmatrix} = \boldsymbol{LU}$$

[83]

$$= \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ 0.1 & 0.1 & 0.1 & 0.1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & & & 0.1 \\ & 1 & & & 0.1 \\ & & 1 & & 0.1 \\ & & & 1 & 0.1 \\ & & & & 0.96 \end{bmatrix}.$$

根据A的非零元所形成的图案，A被称为箭形矩阵. 注意用 GEPP 时A的零元没有被填补(fill in)，所以L和U一起可存放在与A的非零元同一个空间中. 还有，如果计算实质的算术运算次数，即不计用零的乘法或加零，它们只有 12 次[4 次除法计算L的最后行和 8 次乘法和加法更新(5.5)元素]，而不是$\frac{2}{3}n^3 \approx 83$次. 更一般地，若$A$是一个$n \times n$箭形矩阵，它只取$3n-2$个位置而不是$n^2$去存储，$3n-3$次而不是$\frac{2}{3}n^3$次浮点运算来执行高斯消元法. 当$n$是巨大的数时，与稠密阵相比较空间和运算量两者都变得很小.

假如我们已知A'而不是A，A'的行和列的次序与A相反. 这等于线性方程组$Ax=b$中颠倒方程的次序和未知量的次序. 应用 GEPP 于A'时不再置换行，并算到二位小数，得到

$$A' = \begin{bmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 1 & & & \\ 0.1 & & 1 & & \\ 0.1 & & & 1 & \\ 0.1 & & & & 1 \end{bmatrix} = L'U'$$

$$= \begin{bmatrix} 1 & & & & \\ 0.1 & 1 & & & \\ 0.1 & -0.01 & 1 & & \\ 0.1 & -0.01 & -0.01 & 1 & \\ 0.1 & -0.01 & -0.01 & -0.01 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 \\ & 0.99 & -0.01 & -0.01 & -0.01 \\ & & 0.99 & -0.01 & -0.01 \\ & & & 0.99 & -0.01 \\ & & & & 0.99 \end{bmatrix}.$$

现在看到L'和U'已经完全被填补，需要n^2个存储单元. 的确，在算法的第 1 步以后A'的所有非零元已经被填补，所以必须像稠密阵高斯消元法那样做同样的工作$\frac{2}{3}n^3$次.

这说明行和列的次序对节省存储量和工作量极为重要. 即使我们不必担心为数值稳定性忙于选主元(诸如在楚列斯基分解中那样)，选择行和列的最佳置换去极小化存储量或工作量也是一个极其困难的问题. 事实上，它是一个 NP - 完全问题[111]，这意味着所有已知的求最佳置换的算法运行时间关于n指数增长，它甚至比巨大的n的稠密阵高斯消元法代价大得多. 因而我们必须满足于利用几个成功的候选方法的启发. 下面说明其中某些做法：

除了考虑到选择一个好的行和列置换的复杂情况外，还存在稀疏高斯消元法或楚列斯基分解比它们相应的稠密情况更加错综复杂的其他的原因. 首先，需要设计

84

一个只保存 $\boldsymbol{A}$ 的非零元的数据结构；有几个公共的用途[93]. 其次，需要一个数据结构去容纳消元期间填补 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的新元素. 这意味着不是数据结构必须在算法期间动态地增长，就是不费力气地事先计算它而不实际上执行消元. 最后，必须利用数据结构，仅仅执行最少的浮点运算次数，至多与整数和逻辑运算次数相当. 换言之，我们不能承受做 $O(n^3)$ 次整数和逻辑运算，我们想要的是只需很少次数的浮点运算. 这些算法更完整的讨论已超出本书的范围[114,93]，但我们将指出可以利用的软件.

例2.9　用一个更实际的例子来说明楚列斯基分解，这个例子模拟外力造成一个力学结构的位移. 图2-9表示一个具有两个孔的力学结构的二维切片的简单网格. 数学问题是计算网格的所有格子点的位移(它对结构来说是内部的)，网格受到某个实施在结构边界上的力. 从1到 $n=483$ 编号网格点. 实际中的问题具有更大的 n 值. 对力的位移建立的方程是一个线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$，483个网格点中每一个有一行和一列，并且 $a_{ij}\neq 0$ 当且仅当网格点 i 用一线段连接到网格点 j. 这意味着 $\boldsymbol{A}$ 是一个对称矩阵. 它也可证明是正定的，故可使用楚列斯基分解求解 $\boldsymbol{Ax}=\boldsymbol{b}$. 注意 $\boldsymbol{A}$ 只有可能的 $483^2=233\ 289$ 个非零元中的 $nz=3971$ 个非零元，故 $\boldsymbol{A}$ 仅仅有 $3971/233\ 289=1.7\%$ 被填补.（类似的力学建模问题见例4.1和例5.1，那里将详尽地导出矩阵 $\boldsymbol{A}$.）

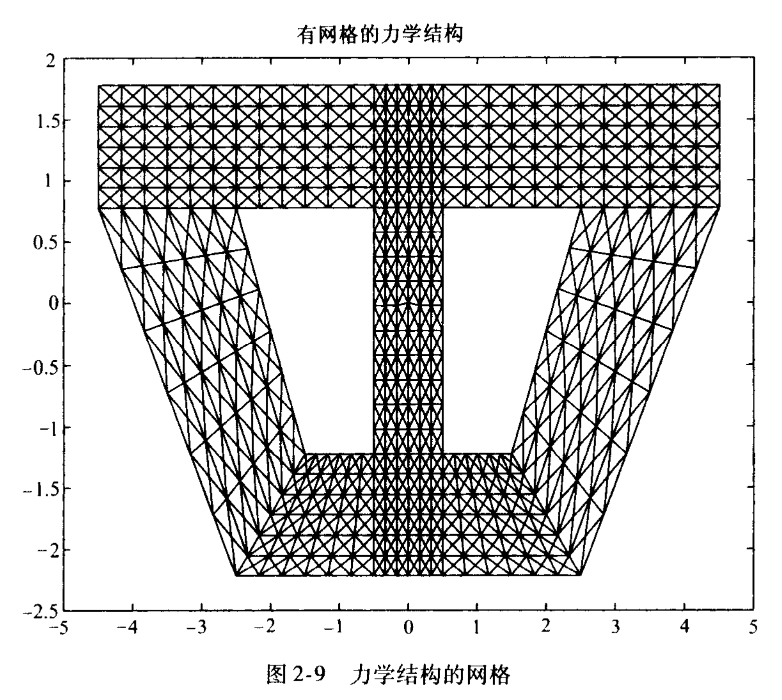


图2-9　力学结构的网格

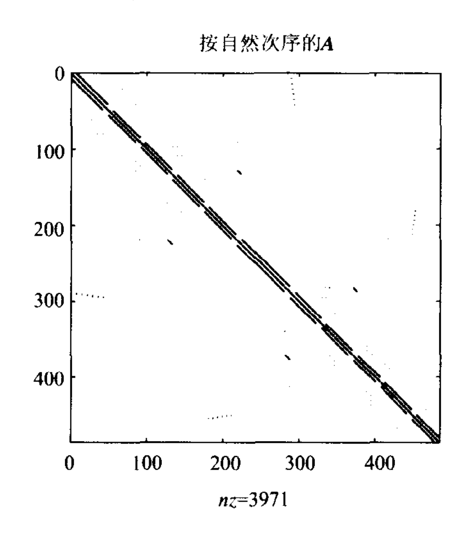
图2-10(见彩插)表示同样的网格(上图)连同矩阵 $\boldsymbol{A}$ 的非零元的图案(下图)，其

中 483 个节点按“自然的”方式排序，逻辑上矩形子结构按行编号，一个子结构接着另一个子结构. 每个这样的子结构的边有一种共同的颜色，这些颜色匹配矩阵中非零元的颜色. 每个子结构有一个标志“$(i:j)$”指出它对应于 $\boldsymbol{A}$ 的 i 行到 j 行和 i 列到 j 列. 对应的子矩阵 $\boldsymbol{A}(i:j,i:j)$ 是一个箭形带状阵. (例2.8 和6.3 节描述其他的情况，其中一个网格导致一个带状矩阵.)连接不同子结构的边是红色的，并且对应于 $\boldsymbol{A}$ 的红色元素. 这些元素离 $\boldsymbol{A}$ 的对角线最远.

标示于图 2-11 中最上面的一对图再次显示 $\boldsymbol{A}$ 以自然次序排序的稀疏结构，以及它的楚列斯基因子 $\boldsymbol{L}$ 的稀疏结构. 对应于 $\boldsymbol{A}$ 的非零元，$\boldsymbol{L}$ 的非零元中是右图中的黑色阴影，$\boldsymbol{L}$ 的新的非零元称为填补(fill-in). $\boldsymbol{L}$ 有 11 533 个非零元，比 $\boldsymbol{A}$ 的下三角形部分的 5 倍还多. 用楚列斯基分解计算 $\boldsymbol{L}$ 的代价仅仅为 296 923 flops，正好是稠密楚列斯基分解所需的 $\frac{1}{3}n^3=3.76\cdot10^7$ flops 的 0.8%.

$\boldsymbol{L}$ 中的非零元数和计算 $\boldsymbol{L}$ 所需的 flops 数可以通过 $\boldsymbol{A}$ 的行和列重新排序而被较大地改变. 图 2-11 中间的一对图说明一种流行的重新排序的结果，这种排序称为 reverse Cuthill-McKee[114,93]，它被设计成使 $\boldsymbol{A}$ 为狭幅的带状阵. 正如所见，对这种情况它是十分成功的，减少 $\boldsymbol{L}$21% 的填补量(从 11 533 减少到 9073)并几乎减少 flop 计算 39%(从 296 923 减少到 181 525).

另一个流行的排序算法称为*极小次数排序*[114,93]，它被设计成在楚列斯基分解的每步中尽可能产生小的填补. 这个结果标示于图 2-11 中最下面的一对图中:$\boldsymbol{L}$ 的填补进一步减少 7%(从 9073 减少到 8440)，但是 flop 计算却增加 9%(从181 525增加到 198 236). ◇



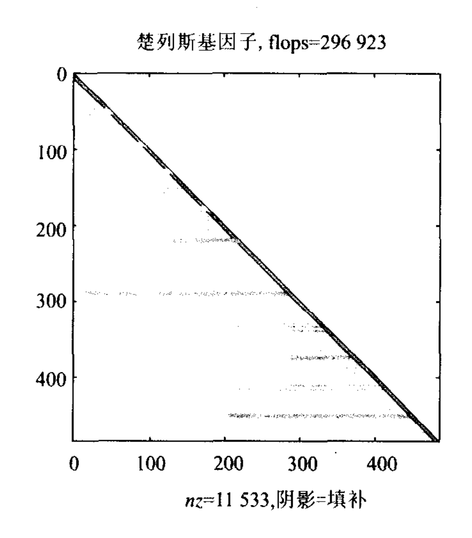


图 2-11 关于各种排序 A 的稀疏性和 flop 数

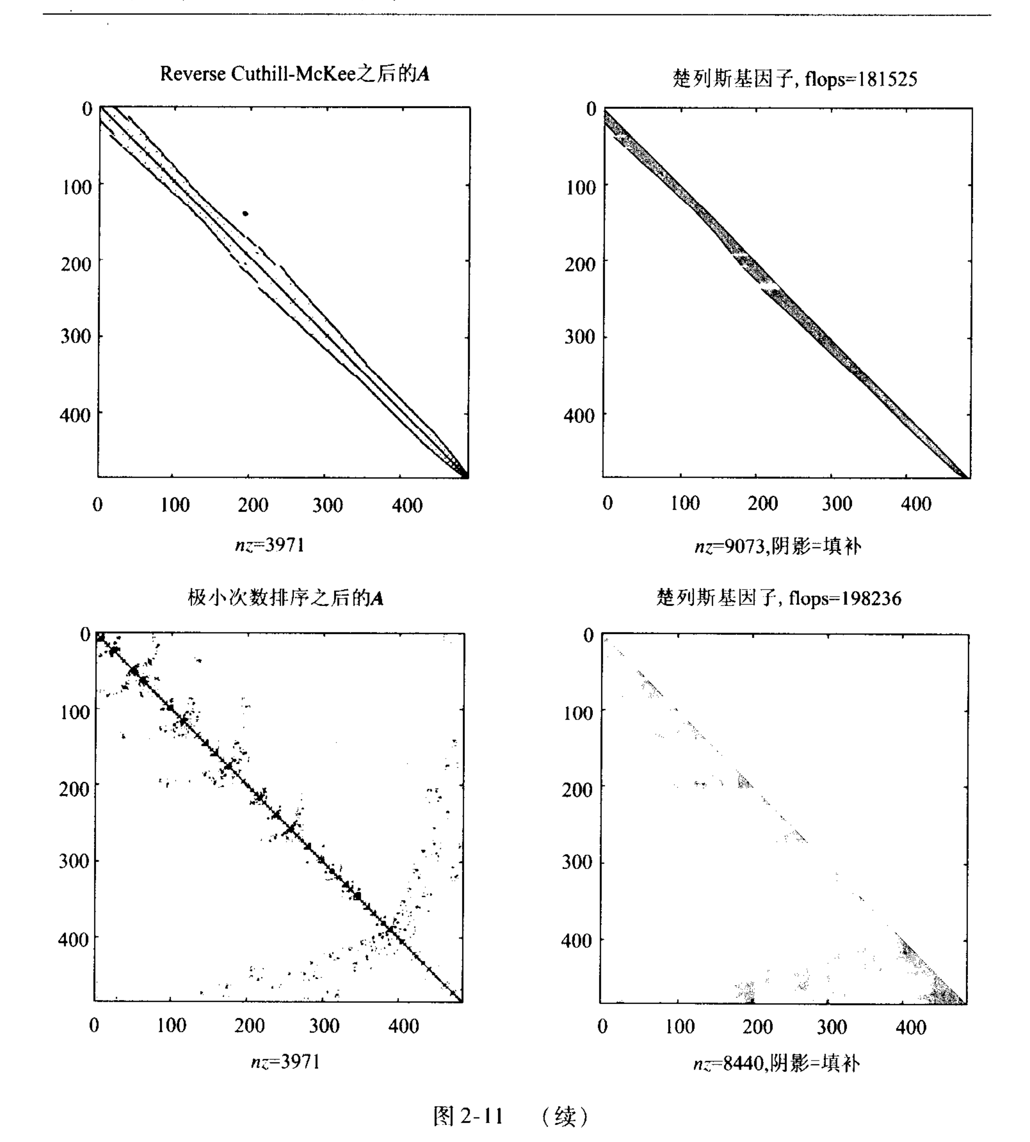
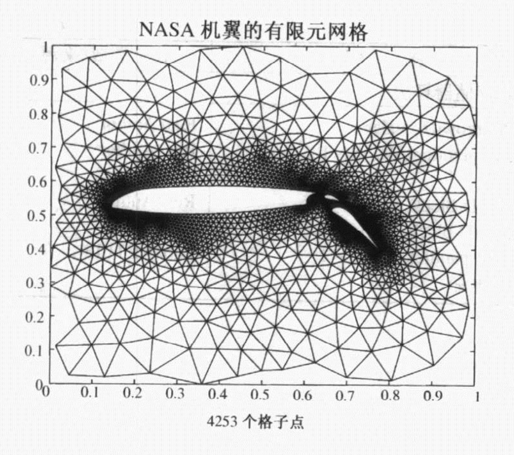


图 2-11　（续）

在 Matlab 中有许多内置的稀疏矩阵的演示(demo)例子，Matlab 中也有许多内置的稀疏矩阵运算(在 Matlab 中输入 "help sparfun" 会列出一个清单)．为了观察这些例子，在 Matlab 中输入 "demo"，点击 "continue"，再点击 "Matlab/Visit"，然后点击 "Matrices/Select a demo/Sparse" 或 "Matrices/Select a demo/Cmd line demos"．例如，图 2-12 表示环绕机翼的网格的例子，其中的目标是计算网格点上环绕机翼的气流．气流对应的偏微分方程导致一个非对称的线性方程组，其稀疏性模式也被展示．

85 ~ 88



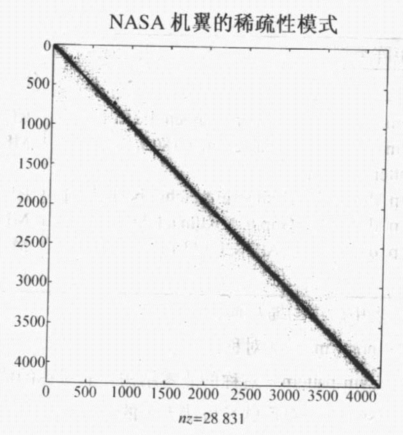


图 2-12 环绕 NASA 机翼的网格

稀疏矩阵软件

除了 Matlab 之外，还有许多用 Fortran 和 C 语言编写的免费和商业稀疏矩阵软件. 因为这仍然是一个活跃的研究领域(尤其对于高性能机器)，所以不可能只推荐某个最佳算法. 表 2-2[177]给出几种候选的可以利用的软件目录. 我们限定自己支持既有代码(或免费的或商业的)，当问题或机器的类型无其他的软件可利用时使用研究型代码. 更完整的目录和下面算法的说明请查阅[177,94].

表 2-2 用直接法解稀疏线性方程组的软件

矩阵类型	名 称	算 法	软件类型/来源
		串行算法	
nonsym.	SuperLU	LL, partial, BLAS - 2.5	Pub/NETLIB
nonsym.	UMFPACK [62,63] MA38(同 UMFPACK)	MF, Markowitz, BLAS - 3	Pub/NETLIB Com/HSL
nonsym.	MA48 [96]	Anal: RL, Markowitz Fact: LL, partial, BLAS - 1, SD	Com/HSL
nonsym.	SPARSE[167]	RL, Markowitz, scalar	Pub/NETLIB
sym - pattern	MUPS[5]	MF, threshold, BLAS - 3	Com/HSL
	MA42 [98]	Frontal, BLAS - 3	Com/HSL
sym.	MA27 [97]/MA47[95]	MF, LDL^T, BLAS - 1/BLAS - 3	Com/HSL
s. p. d.	Ng & Peyton[191]	LL, BLAS - 3	Pub/Author
		共享存储器算法	
nonsym.	SuperLU	LL, partial, BLAS - 2.5	Pub/UCB
nonsym.	PARASPAR [270,271]	RL, Markowitz, BLAS - 1, SD	Res/Author
sym - pattern	MUPS [6]	MF, threshold, BLAS - 3	Res/Author
nonsym.	George & Ng[115]	RL, partial, BLAS - 1	Res/Author
s. p. d.	Gupta et al. [133]	LL, BLAS - 3	Com/SGI Pub/Author
s. p. d.	SPLASH [155]	RL, 2 - D block, BLAS - 3	Pub/Stanford

（续）

矩阵类型	名　称	算　法	软件类型/来源
		分布式存储器算法	
sym.	van der Stappen [245]	RL, Markowitz, scalar	Res/Author
sym - pattern	Lucas et al. [180]	MF, no pivoting, BLAS - 1	Res/Author
s. p. d.	Rothberg & Schreiber[207]	RL, 2 - D block, BLAS - 3	Res/Author
s. p. d.	Gupta & Kumar[132]	MF, 2 - D block, BLAS - 3	Res/Author
s. p. d.	CAPSS [143]	MF, full parallel, BLAS - 1（需要坐标）	Pub/NETLIB

表中所用的缩写词：

nonsym. = 非对称的.

sym-pattern = 对称的非零结构，非对称的值.

sym. = 对称的且可能不定的.

s. p. d. = 对称的且正定的.

MF，LL 和 RL = 多面，左看和右看.

SD = 对相当稠密的尾部子阵算法转换到稠密的代码.

Pub = 可公开得到的，作者们可帮助使用代码.

Res = 文献中已出版的但不可能从作者们那里得到.

Com = 商业的.

HSL = Harwell 子程序库，网址为 http://www.rl.ac.uk/departments/ccd/numerical/hsl/hsl.html.

UCB = http://www.cs.berkeley.edu/~xiaoye/superlu.html.

Stanford = http://www-flash.stanford.edu/apps/SPLASH/.

表 2-2 如下安排：顶部标志串行算法(serial algorithm)的程序组是为单处理器工作站和个人计算机设计的. 共享存储器算法(shared-memory algorithm)是为诸如 Sun SPARCcenter 2000 [238]，SGI Power Challenge [223]，DEC AlphaServer 8400 [103]和 Cray C90/J90[253,254]对称多处理机设计的分布式存储器算法(distributed-memory algorithm)是为诸如 IBM SP-2[256]，Intel Paragon[257]，Cray T3 系列[255]以及工作站网络[9]设计的. 正如你所见，已编写的大部分的软件是用于串行机的，有些是用于共享存储器机器的，极少数(研究软件除外)是用于分布式存储器机器的.

第一列为矩阵类型. 可能包括非对称、对称模式(即或者 $a_{ij}=a_{ji}\neq 0$ 或者两者可能非零但不相等)、对称(也许不定的)，以及对称正定(s. p. d.). 第二列为程序名或作者名.

第三列为算法的某些细节，确实比课文中已经详尽地说明过的更多：LL(左看)、RL(右看)、frontal(前面)、MF(多面)和 LDL^T 指的是用不同方式组织确定高斯消元法的三个嵌套循环. Partial(部分的)、Markowitz 和 threshold(阈值)指的是不同的选主元方法. 2-D block 指的是并行处理机对矩阵的哪部分负责. CAPSS 假定线性方程组是由格子点确定的，为了在处理机中分配矩阵需要格子点的 x，y 和 z 坐标.

第三列也描述最内层循环的组织，它可以是 BLAS1，BLAS2，BLAS3 或标量. SD 指的是当尾部$(n-k)\times(n-k)$子阵相当稠密时算法转换到稠密高斯消元法.

第四列描述软件的类型和可行性，包括它是免费的还是商业的，以及如何得到它.

2.7.5 不超过 $O(n^2)$ 个参数的稠密矩阵

90 ~ 91

这是一个包罗万象的标题，它包括很多种实际问题中产生的矩阵. 我们只提及少数情况. 范特蒙德矩阵(Vandermonde matrix)具有形式

$$V=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & & x_n \\ x_0^2 & x_1^2 & & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_0^{n-1} & x_1^{n-1} & & x_n^{n-1} \end{bmatrix}.$$

注意矩阵 - 向量乘法

$$V^{\mathrm{T}}\cdot[a_0,\cdots,a_n]^{\mathrm{T}}=\left[\sum a_i x_0^i,\cdots,\sum a_i x_n^i\right]^{\mathrm{T}}$$

等价于多项式求值. 所以求解 $V^{\mathrm{T}}a=y$ 是多项式插值. 利用牛顿插值公式可以以$\frac{5}{2}n^2$ flops 次而不是$\frac{2}{3}n^3$ flops 求解$V^{\mathrm{T}}a=y$. 还有一个类似的决窍以$\frac{5}{2}n^2$ flops 求解 $Va=y$. 见[121,p. 178].

柯西矩阵 C 有元素

$$c_{ij}=\frac{\alpha_i\beta_j}{\xi_i-\eta_j}$$

其中$\boldsymbol{\alpha}=[\alpha_1,\cdots,\alpha_n]$,$\boldsymbol{\beta}=[\beta_1,\cdots,\beta_n]$,$\boldsymbol{\xi}=[\xi_1,\cdots,\xi_n]$和$\boldsymbol{\eta}=[\eta_1,\cdots,\eta_n]$是已知的向量. 最有名的例子是病态希尔伯特矩阵 H，$h_{ij}=1/(i+j-1)$. 这些矩阵出现在用有理函数插值数据时. 假定要求具有固定极点 η_j 的有理函数的系数 x_j，

$$f(z)=\sum_{j=1}^{n}\frac{x_j}{z-\eta_j}$$

使得$f(\xi_i)=y_i,i=1$ 到 n. 把这 n 个方程$f(\xi_i)=y_i$ 合在一起构成一个系数阵是柯西矩阵的 $n\times n$ 线性方程组. 柯西矩阵之逆是柯西矩阵，基于C^{-1}与插值的关系，C^{-1}有一个闭型表达式：

$$(C^{-1})_{ij}=\beta_i^{-1}\alpha_j^{-1}(\xi_j-\eta_i)P_j(\eta_i)Q_i(-\xi_j),$$

其中 $P_j(\cdot)$ 和 $Q_i(\cdot)$ 是拉格朗日插值多项式

$$P_j(z)=\prod_{k\neq j}\frac{\xi_k-z}{\xi_k-\xi_j},\ Q_i(z)=\prod_{k\neq i}\frac{-\eta_k-z}{-\eta_k+\eta_i}.$$

特普利茨矩阵(Toeplitz matrix)外表特征是

$$\begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_n \\ a_{-1} & \ddots & \ddots & \ddots & \vdots \\ a_{-2} & \ddots & \ddots & \ddots & a_2 \\ \vdots & \ddots & \ddots & \ddots & a_1 \\ a_{-n} & \cdots & a_{-2} & a_{-1} & a_0 \end{bmatrix};$$

92

即它们沿着对角线是常数. 它们产生于信号处理问题中. 存在只用 $O(n^2)$ 次运算求解这种方程组的算法.

所有这些方法推广到只有 $O(n)$ 个参数的许多其他类似的矩阵. 最新的综述可见[121,p. 183]或[160].

2.8 第2章的参考书目和其他的话题

有关求解线性方程更多的细节一般可在[121]的第3和第4章中找到. 在[71]中进一步研究了条件数和最接近不适定问题距离之间的倒数关系. 主元增长的正常情况分析在[242]中描述，完全选主元的坏主元增长的一个例子在[122]中给出. 条件估计量在[138,146,148]中描述. 单精度迭代精化在[14,225,226]中分析. 有关线性方程解算器误差分析的全面的讨论可在[149]中找到，它包括大部分这些话题.

关于对称不定分解，见[44]. 稀疏矩阵算法在[114,93]以及表2-2中众多的参考书目中描述. 本章中所描述的有关稠密和带状阵的许多算法执行过程在LAPACK和CLAPACK[10]中可以得到，它包括适用于高性能计算机块算法的一个讨论. 并行执行过程在ScaLAPACK[34]中可以得到. BLAS在[87,89,169]中描述. 这些程序和另一些程序可以在NETLIB中找到. 矩阵乘法分块策略的分析在[151]中给出. Strassen矩阵乘法算法在[3]中提出，它的实际性能在[22]中描述，而其数值稳定性在[77,149]中描述. 并行算法和其他块算法的综述在[76]中给出. 仅仅和 $O(n)$ 个参数有关的结构稠密矩阵算法最新的综述是[160]. 更多的关于稀疏直接法的材料，见[93,94,114,177].

2.9 第2章问题

问题2.1(容易) 利用你喜欢的万维网浏览器转到NETLIB(http://www.netlib.org)，并回答下列问题

1. 你需要一个用双精度计算实对称矩阵的特征值和特征向量的Fortran子程序. 利用搜索工具在NETLIB库中找一个. 报告子程序的名称及URL以及你如何找到它的.

93

2. 利用性能数据库服务器(Performance Database Server)求出用高斯消元法解100×100稠密线性方程组的当今世界速度记录. 用Mflops的速度是什么，哪台机器达

到这个速度？对 1000×1000 稠密线性方程组以及“像你所要求那样大的”稠密线性方程组做同样的工作. 利用同样的数据库，求出你的工作站解 100×100 稠密线性方程组能有多快. 提示：着眼于 LINPACK 基准.

问题 2.2(容易) 考虑解 $\boldsymbol{AX}=\boldsymbol{B}$，其中 $\boldsymbol{A}$ 是 $n\times n$ 阶矩阵而 $\boldsymbol{X}$ 和 $\boldsymbol{B}$ 是 $n\times m$ 阶矩阵. 有两个显而易见的算法. 第一个算法利用高斯消元法分解 $\boldsymbol{A}=\boldsymbol{PLU}$，然后用向前和向后回代求解 $\boldsymbol{X}$ 的每列. 第二个算法利用高斯消元法计算 $\boldsymbol{A}^{-1}$，然后相乘 $\boldsymbol{X}=\boldsymbol{A}^{-1}\boldsymbol{B}$. 计算每个算法所需要的 flops 数，并证明第一个算法需要较少的 flops.

问题 2.3(中等) 设 $\|\cdot\|$ 是 2-范数. 给定一个非奇异矩阵 $\boldsymbol{A}$ 和一个向量 $\boldsymbol{b}$，证明对于充分小的 $\|\delta\boldsymbol{A}\|$，存在非零的 $\delta\boldsymbol{A}$ 和 $\delta\boldsymbol{b}$ 使得不等式(2.2)为一个等式. 这证明了称 $\kappa(\boldsymbol{A})=\|\boldsymbol{A}^{-1}\|\cdot\|\boldsymbol{A}\|$ 为 $\boldsymbol{A}$ 的条件数是有道理的. 提示：利用定理 2.1 证明中的思想.

问题 2.4(困难) 证明界(2.7)和(2.8)是可以达到的.

问题 2.5(中等) 证明定理 2.3. 给定残差 $\boldsymbol{r}=\boldsymbol{A}\hat{\boldsymbol{x}}-\boldsymbol{b}$，利用定理 2.3 证明界(2.9)不大于界(2.7). 这就说明为什么如 2.4.4 节中描述的那样 LAPACK 计算基于(2.9)的界.

问题 2.6(容易) 证明引理 2.2.

问题 2.7(容易. Z. Bai) 若 $\boldsymbol{A}$ 是一个非奇异的对称阵且有分解 $\boldsymbol{A}=\boldsymbol{LDM}^{\mathrm{T}}$，其中 $\boldsymbol{L}$ 和 $\boldsymbol{M}$ 是单位下三角形阵而 $\boldsymbol{D}$ 是对角阵. 证明 $\boldsymbol{L}=\boldsymbol{M}$.

问题 2.8(困难) 考虑求解下列 2×2 线性方程组的两个方法：

$$\boldsymbol{A}x=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}b_1\\ b_2\end{bmatrix}=\boldsymbol{b}.$$

算法 1 部分选主元高斯消元法(GEPP).

算法 2 克拉默法则：

$$\begin{aligned}\det&=a_{11}*a_{22}-a_{12}*a_{21},\\ x_1&=(a_{22}*b_1-a_{12}*b_2)/\det,\\ x_2&=(-a_{21}*b_1+a_{11}*b_2)/\det.\end{aligned}$$

94

利用数值例子说明克拉默法则不是向后稳定的. 提示：选择几乎奇异的矩阵并且 $[b_1,b_2]^{\mathrm{T}}\approx[a_{12},a_{22}]^{\mathrm{T}}$. 向后稳定性如何暗示残差的大小？数值例子可在纸上手算(例如，用4位浮点小数)，在一台计算机上，或用掌上计算器来做.

问题 2.9(中等) 设 $\boldsymbol{B}$ 是 $n\times n$ 上双对角阵，即非零元仅在主对角线和第一上对角线上. 导出一个精确地(不计舍入)计算 $\kappa_\infty(\boldsymbol{B})\equiv\|\boldsymbol{B}\|_\infty\|\boldsymbol{B}^{-1}\|_\infty$ 的算法. 换言之，你不应该利用像 Hager 估计量那样的迭代算法. 你的算法应该尽可能代价小. 使用不超过 $2n-2$ 次加法，n 次乘法，n 次除法，$4n-2$ 个绝对值和 $2n-2$ 次比较来做应该是可能的.(接近于这个估计的任何算法是令人满意的).

问题 2.10(容易. Z. Bai) 设 $\boldsymbol{A}$ 是 $n\times m$ 阶矩阵. $n\geq m$，证明 $\|\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\|_2=\|\boldsymbol{A}\|_2^2$，以及 $\kappa_2(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})=\kappa_2(\boldsymbol{A})^2$.

设 M 是 $n\times n$ 阶正定阵而 L 是它的楚列斯基因子使得 $M=LL^{T}$. 证明 $\|M\|_2=\|L\|_2^2$ 以及 $\kappa_2(M)=\kappa_2(L)^2$.

问题2.11(容易. Z. Bai)　设 A 对称正定. 证明 $|a_{ij}|<(a_{ii}a_{jj})^{1/2}$.

问题2.12(容易. Z. Bai)　证明：若

$$Y=\begin{pmatrix} I & Z \\ 0 & I \end{pmatrix},$$

其中 I 是 $n\times n$ 阶单位阵，则 $\kappa_F(Y)=\|Y\|_F\|Y^{-1}\|_F=2n+\|Z\|_F^2$.

问题2.13(中等)　在本题中给定一个求解 $Ax=b$ 的快速方法，要问如何求解 $By=c$，其中 $A-B$ 在某种意义下是“小”的.

1. 证明 Sherman - Morrison 公式：设 A 非奇异. u 和 v 是列向量且 $A+uv^{T}$ 非奇异. 则 $(A+uv^{T})^{-1}=A^{-1}-(A^{-1}uv^{T}A^{-1})/(1+v^{T}A^{-1}u)$.

更一般地，证明 Sherman-Morrison-Woodbury 公式：设 U 和 V 是 $n\times k$ 阶长方阵，其中 $k\leqslant n$，而 A 是 $n\times n$ 阶矩阵. 则 $T=I+V^{T}A^{-1}U$ 非奇异当且仅当 $A+UV^{T}$ 非奇异，此时 $(A+UV^{T})^{-1}=A^{-1}-A^{-1}UT^{-1}V^{T}A^{-1}$.

2. 若你有一个求解 $Ax=b$ 的快速算法，证明如何构造 $By=c$ 的一个快速解算器，其中 $B=A+uv^{T}$.

3. 假如 $\|A-B\|$ 是“小的”并有一个解 $Ax=b$ 的快速算法. 描述一个求解 $By=c$ 的迭代格式. 你预期你的算法收敛多快？提示：利用迭代精化.

95

问题2.14(中等，程序设计)　利用 Netlib 得到用部分选主元高斯消元法求解 $Ax=b$ 的子程序. 应该从 LAPACK(Fortran 语言，NETLIB/lapack)或者从 CLAPACK(C 语言，NETLIB/clapack)得到它. sgesvx 是两种情况中的主程序(也有你可能想要的较简单的程序 sgesv). 修正的 sgesvx(以及它可能调用的其他子程序)执行完全选主元而不是部分选主元. 调用这个新的程序 gecp. 修改 sgetf2 并使用它代替 sgetrf 可能是最简单的. 一个 Matlab 实现见 HOMEPAGE/Matlab/gecp. m. 对许多随机生成的30阶左右的矩阵测试 sgesvx 和 gecp. 选择 x 并构成 $b=Ax$，你可以使用已知正确解的例子. 检查计算解 $\hat{x}$ 的精度如下：首先，分析用软件获得的误差界 FERR(向前误差)和 BERR(向后误差)；按你自己的言语说明这些界是什么意思. 利用正确解的知识证实 FERR 是正确的. 其次，用明确地求逆矩阵计算精确的条件数，并把它与用软件获得的估计 RCOND 比较. (实际上，RCOND 是条件数倒数的一个估计.)第三，确认 $\frac{\|\hat{x}-x\|}{\|\hat{x}\|}$ 是以 macheps/RCOND 适当的倍数为界. 第四，应该证实在每种情况(尺度的)向后误差 $R\equiv\|A\hat{x}-b\|/((\|A\|\cdot\|x\|+\|b\|)\cdot \text{macheps})$ 的阶为1.

更准确地说，你的解答应包括 gecp 的一份良好文档的程序清单，以及所生成的那几个随机矩阵的说明(见下面)，还有包括下面的一些列的表(最好每个数据列对照第一列画出的图像)：

- 测试矩阵数(说明如何在生成中识别它)；

- 它的维数；
- 从 sgesvx：
  - —由代码获得主元素增长因子（理论上这个因子不比1大多少）；
  - —它的估计条件数（1/RCOND）；
  - —1/RCOND 与明确地算出的条件数之比（理论上这个比应该接近于1）；
  - —误差界 FERR；
  - —FERR 与真正的误差之比（理论上这个比应该至少是1但不会大许多，除非你是“幸运的”，且真正的误差为0）；
  - —真正的误差与 ε/RCOND 之比（理论上这个比应该至多是1或少一点点，除非你是“幸运的”，且真正的误差为0）；
  - —尺度的向后误差 R/ε[理论上这个值应该是 $O(1)$ 或者可能是 $O(n)$]；
  - —向后误差 BERR/ε[理论上这个值应该是 $O(1)$ 或者可能是 $O(n)$]；
  - —按秒计算的运行时间；
- 像对 sgesvx 那样相同的数据给 gecp.

96

只需要打印数据到一个小数位，因为我们只关心近似的量级. 误差界真实地界定误差吗？如何比较 sgesvx 和 gecp 的速度？

对许多方程组得到精确的记时是困难的，因为许多计时器分辨率较低，故应该如下计算运行时间：

```
t_1 = 迄今为止的时间
for i = 1 to m
    建立问题
    解决问题
end for
t_2 = 迄今为止的时间
for i = 1 to m
    建立问题
end for
t_3 = 迄今为止的时间
t = ((t_2 - t_1) - (t_3 - t_2))/m
```

m 应该选得足够大，使得 $t_2 - t_1$ 至少是几秒钟. 因而 T 应该是解决问题的一个可靠的时间估计.

你应该测试若干良态问题和若干病态问题. 为生成一个良态矩阵，设 $\boldsymbol{P}$ 是一个置换阵，并在每个元素上加一个小的随机数. 为生成一个病态矩阵，设 $\boldsymbol{L}$ 是一个具有微小对角元和适中的次对角元的随机下三角形阵. 设 $\boldsymbol{U}$ 是一个类似的上三角形阵，并设 $\boldsymbol{A} = \boldsymbol{LU}$.（还有一个 LAPACK 子程序 slatms 生成具有给定条件数的随机矩阵，如果你喜欢的话可以使用它.）

同样对下列 $n\times n$ 阶矩阵类，$n=1$ 直到30(若按双精度运行，你可能需要运行直到 $n=60$)，测试两个解算器. 这里仅指出 $n=5$ 情况，其余情况是类似的：

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}.$$

根据2.4节中的误差分析说明结果的精确性.

97

你的解答应该不包含任何矩阵元素表或解分量.

除了讲授误差界外，本题的一个目的是让你知道设计良好的数值软件是怎样的. 实际上，人们常常使用或修改现存的软件，而不是亲自从头开始编写新软件.

问题2.15(中等；程序设计)　这个问题与问题2.14有关，编写 sgesvx 的另一个版本，称为 sgesvxdouble 这个程序在迭代精化期间按双精度计算残差. 修正 sgesvx 中的误差界 FERR 来反映这个改进的精度. 说明你的修正. (可能需要你首先说明 sgesvx 如何计算它的误差界). 对上题中同样的一组例子产生一个类似的数据表. 何时 sgesvxdouble 比 sgesvx 更精确?

问题2.16(困难)　说明如何改造楚列斯基算法(算法2.11)，利用3级 BLAS 做算法的大部分运算. 仿效算法2.10.

问题2.17(容易)　假如在 Matlab 中有一个 $n\times n$ 阶矩阵 $\boldsymbol{A}$ 和一个 $n\times 1$ 阶矩阵 $\boldsymbol{b}$. 在 Matlab 中，$\boldsymbol{A}\backslash\boldsymbol{b}$，$\boldsymbol{b}'/\boldsymbol{A}$ 和 $\boldsymbol{A}/\boldsymbol{b}$ 意指什么? $\boldsymbol{A}\backslash\boldsymbol{b}$ 如何不同于 $\mathrm{inv}(\boldsymbol{A})*\boldsymbol{b}$?

问题2.18(中等的)　设

$$\boldsymbol{A}=\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix},$$

其中 $\boldsymbol{A}_{11}$ 是 $k\times k$ 阶非奇异矩阵. 则 $\boldsymbol{S}=\boldsymbol{A}_{22}-\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}$ 称为 $\boldsymbol{A}$ 中 $\boldsymbol{A}_{11}$ 的舒尔补，或简称舒尔补.

1. 证明不选主元高斯消元法 k 步之后 $\boldsymbol{A}_{22}$ 已被 $\boldsymbol{S}$ 覆盖.

2. 假如 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$，$\boldsymbol{A}_{11}$ 正定而 $\boldsymbol{A}_{22}$ 负定($-\boldsymbol{A}_{22}$ 正定). 证明 $\boldsymbol{A}$ 是非奇异的，不选主元高斯消元法将以精确的算术运算工作，但(使用 2×2 例子)不选主元高斯消元法可能数值不稳定.

问题2.19(中等)　矩阵 $\boldsymbol{A}$ 称为严格列对角占优的，或简称对角占优的，若

$$|a_{ii}|>\sum_{j=1,j\neq i}^{n}|a_{ji}|.$$

98

- 证明 $\boldsymbol{A}$ 非奇异. 提示：利用 Gershgorin 定理.
- 证明部分选主元高斯消元法实际上不置换任何行，即它等同于不选主元的高斯消元法.

　　提示：证明高斯消元法一步之后. 尾部 $(n-1)\times(n-1)$ 子阵，$\boldsymbol{A}$ 中 a_{11} 的

舒尔补仍旧是对角占优的(舒尔补更多的讨论见问题2.18).

问题2.20(容易, Z. Bai) 给定一个 $n \times n$ 阶非奇异阵 $\boldsymbol{A}$, 利用部分选主元的高斯消元法如何有效地求解下列问题?

(a) 解线性方程组 $\boldsymbol{A}^k\boldsymbol{x} = \boldsymbol{b}$, 其中 k 是一个正整数.

(b) 计算 $\alpha = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{b}$.

(c) 解矩阵方程 $\boldsymbol{AX} = \boldsymbol{B}$, 其中 $\boldsymbol{B}$ 是 $n \times m$ 阶矩阵.

你应该 (1) 描述你的算法; (2) 以伪代码呈现它们(使用 Matlab 类语言, 不应该写下 GEPP 算法); (3) 给出所需的 flops.

问题2.21(中等) 证明: Strassen 算法(算法2.8)正确地相乘 $n \times n$ 阶矩阵, 其中 n 是2的幂.

99

第3章 线性最小二乘问题

3.1 概 述

给定 $m\times n$ 阶矩阵 $\boldsymbol{A}$ 和 $m\times 1$ 向量 $\boldsymbol{b}$，线性最小二乘问题是求一个 $n\times 1$ 向量 $\boldsymbol{x}$ 使 $\|\boldsymbol{Ax}-\boldsymbol{b}\|_2$ 达到极小. 若 $m=n$ 且 $\boldsymbol{A}$ 非奇异，则答案即为 $\boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{b}$. 但当 $m>n$ 即方程个数大于未知量个数时，问题称为*超定的*，一般没有 $\boldsymbol{x}$ 精确地满足 $\boldsymbol{Ax}=\boldsymbol{b}$. 偶然会遇到*亚定的*(underdetermined)问题，其中 $m<n$，但是我们将集中讨论更为普遍的超定的问题.

本章内容组织如下. 本概述的其余部分描述最小二乘问题的三个应用，*曲线拟合*，*噪声数据的统计建模*和*大地测量建模*(geodetic modeling). 3.2 节讨论求解最小二乘问题的三种标准方法：*正规方程*，*QR 分解*以及*奇异值分解*(SVD). 因为在后面几章中经常使用 SVD 作为一个工具，所以我们推导它的几个性质(尽管 SVD 算法留在第5章中讨论). 3.3 节讨论最小二乘问题的扰动理论，3.4 节讨论执引过程细节以及主要算法 QR 分解的舍入误差分析. 舍入误差分析应用于许多使用正交矩阵的算法，包括第4章和第5章中特征值和 SVD 的许多算法. 3.5 节讨论特别病态的秩亏最小二乘问题的情况，以及如何精确地求解它们. 3.7 节以及本章末尾的问题对其他类型的最小二乘问题和稀疏问题的软件给出一点建议.

例 3.1 最小二乘的一个典型的应用是*曲线拟合*. 假如有 m 个数对 $(y_1,b_1),\cdots,(y_m,b_m)$，并且要求找出"最佳的"三次多项式作为 y_i 的函数拟合 b_i. 这意味着求多项式系数 $x_1,\cdots,x_4$ 使得多项式 $p(y)=\sum_{j=1}^{4}x_jy^{j-1}$ 使残差 $r_i\equiv p(y_i)-b_i$ 达到极小，$i=1$ 到 m. 也可把它写成极小化

$$r\equiv\begin{bmatrix}r_1\\r_2\\\vdots\\r_m\end{bmatrix}=\begin{bmatrix}p(y_1)\\p(y_2)\\\vdots\\p(y_m)\end{bmatrix}-\begin{bmatrix}b_1\\b_2\\\vdots\\b_m\end{bmatrix}$$

$$=\begin{bmatrix}1&y_1&y_1^2&y_1^3\\1&y_2&y_2^2&y_2^3\\\vdots&\vdots&\vdots&\vdots\\1&y_m&y_m^2&y_m^3\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}-\begin{bmatrix}b_1\\b_2\\\vdots\\b_m\end{bmatrix}$$

$$\equiv\boldsymbol{A}\cdot\boldsymbol{x}-\boldsymbol{b},$$

其中 r 和 b 是 $m\times 1$ 阶矩阵，A 是 $m\times 4$ 阶矩阵，x 是 4×1 阶矩阵. 为极小化 r，可以选择任何范数，如 $\|r\|_\infty$，$\|r\|_1$，或 $\|r\|_2$. 最后一个范数对应于极小化平方残差之和 $\sum_{i=1}^{m} r_i^2$，这是一个线性最小二乘问题.

图 3-1(见彩插)指出一个例子，用递增次数的多项式在 23 个点 $y=-5,-4.5,-4,\cdots,5.5,6$ 上拟合光滑函数 $b=\sin(\pi y/5)+y/5$. 图 3-1 的左边标出用圆表示的数据点以及 4 个次数为 1，3，6 和 19 的不同的近似多项式. 图 3-1 的右边标出与次数相对的残差范数 $\|r\|_2$，次数从 1 到 20. 注意，当次数从 1 次增加到 17 次时，残差范数是递减的. 我们期待这种特性，因为增加多项式次数将使我们较好地拟合数据.

但是当达到 18 次时，残差范数突然显著地增加. 在图的左边(蓝色的线)可以看出 19 次多项式的图是多么无规律. 正如我们将在后面看到的那样，这是由于病态引起的. 原则上，人们只用相对低次的多项式做多项式拟合以避免病态[61]. 多项式拟合可利用 Matlab 中的函数 polyfit.

下面是另一种多项式拟合. 更一般地，有一组从 $\mathbb{R}^k$ 到 $\mathbb{R}$ 的线性无关函数 $f_1(y),\cdots,f_n(y)$，以及一组点 $(y_1,b_1),\cdots,(y_m,b_m)$，$y_i\in\mathbb{R}^k$，$b_i\in\mathbb{R}$，希望对形如 $b=\sum_{j=1}^{n}x_jf_j(y)$ 的这些点找出一个最佳的拟合. 换言之，要选择 $x=[x_1,\cdots,x_n]^T$ 去极小化残差 $r_i\equiv\sum_{j=1}^{n}x_jf_j(y_i)-b_i, 1\leqslant i\leqslant m$. 设 $a_{ij}=f_j(y_i)$，可以把这个写成 $r=Ax-b$，其中 A 是 $m\times n$ 阶矩阵，x 是 $n\times 1$ 阶矩阵，而 b 和 r 是 $m\times 1$ 阶矩阵. 基函数 $f_i(y)$ 的一种好的选择比利用多项式可导致较好的拟合和较少的病态[33,84,168]. ◇

例 3.2 在统计模型中，人们通常希望根据若干观察来估计某些参数 x_j，其中观察数据被噪声污染了. 例如，假如我们希望根据大学新生的高中成绩平均值(GPA)(a_1)和两次学习能力测试得分，口头的(a_2)和量化的(a_3)，来预测他们的大学入学考试成绩平均值(b)作为大学入学过程的一部分. 根据入学新生过去的数据，可以构造形如 $b=\sum_{j=1}^{3}a_jx_j$ 的一个线性模型. 观察数据是 a_{i1},a_{i2},a_{i3} 和 b_i，把 m 个学生的每组观察数据放置于数据库中. 因而，应该极小化

102

$$r\equiv\begin{bmatrix}r_1\\r_2\\\vdots\\r_m\end{bmatrix}=\begin{bmatrix}a_{11}&a_{12}&a_{13}\\a_{21}&a_{22}&a_{23}\\\vdots&\vdots&\vdots\\a_{m1}&a_{m2}&a_{m3}\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}-\begin{bmatrix}b_1\\b_2\\\vdots\\b_m\end{bmatrix}\equiv A\cdot x-b,$$

这可以作为一个最小二乘问题来处理.

下面是最小二乘的一种统计学的说法，它被统计学家称为*线性回归*：假定 a_i 是精确地已知的，而只有 b 中有噪声. 并且每个 b_i 中的噪声是独立的并具有 0 均值和相同标准差 σ 的正态分布. 设 x 是最小二乘问题的解以及 x_T 是参数的真值. 则 x 称为 x_T 的极大似然估计，并且误差 $x-x_T$ 是正态分布的，在每个分量中具有零均值和协方差阵 $\sigma^2(A^TA)^{-1}$. 下面当我们利用正规方程求解最小二乘问题时将再次看到矩阵

$(A^{\mathrm{T}}A)^{-1}$. 与统计学相关的更多细节[1]，可见[33,259]. ◇

例 3.3 最小二乘问题最早是由高斯在求解法国政府的一个实际问题中提出和明确地表达的. 精确地知道不同的公民私有的小块土地之间的边界具有重要的经济上和法学上的原因. 测量人员应该实地考察并尝试建立这些边界，度量某些角度和距离，然后从已知的界标作三角定位. 随着时光流逝，有必要改进已知界标位置的精度. 故白大外出的测量人员复测许多角度和界标之间的距离，而将解决如何使用这些更准确的测量数据并更新政府的界标位置数据库的任务委派给高斯. 由此他发明了最小二乘法，我们将简要地加以说明[33].

103

委派给高斯解决的问题没有停止，而且必须定时重新调查. 1974 年，美国国家土地测绘局承担更新大约由 700 000 个点组成的美国测量数据库. 该项目已增长到包括为土木工程师和地区规划人员设计建筑项目，以及为地球物理学者研究地壳上建筑物移动(它每年可能移动 5 厘米)提供足够准确的数据. 相应的最小二乘问题是在那个时候曾经求解过的最大的问题：大约为含 400 000 个未知量的 2 500 000 个方程. 它是非常稀疏的，1978 年的计算机已可以处理，计算得以完成.

现在简要地讨论这个问题的明确表达式. 它实际上是非线性的，并且通过一系列线性化逼近它来求解，其中每一个是一个线性最小二乘问题. 数据库由一个点(界标)序列组成，每个点用位置标示：纬度、经度或者海拔高度. 为简单说明起见，假定地球是平坦的，并假定每个点 i 用线性坐标 $z_i=(x_i,y_i)^{\mathrm{T}}$ 标记. 对每个点我们希望计算一个校正 $\delta z_i=(\delta x_i,\delta y_i)^{\mathrm{T}}$ 使得校正的位置 $z'_i=(x'_i,y'_i)^{\mathrm{T}}=z_i+\delta z_i$ 几乎更匹配新的更精确的测量. 这些测量同时包括选取的点和点之间的距离和从点 i 到点 j 以及点 i 到点 k 线段之间的角度(如图 3-2). 为观察这些新的测量如何转换到约束中去，考察图 3-2 中的三角形. 夹角是用它们(校正)的位置标示的，角 θ 及边长 L 也被指明. 由这个数据，利用三角恒等式容易写出约束条件. 例如，一个准确的 θ_i 的测量导致约束条件

$$\cos^2\theta_i=\frac{[(z'_j-z'_i)^{\mathrm{T}}(z'_k-z'_i)]^2}{(z'_j-z'_i)^{\mathrm{T}}(z'_j-z'_i)\cdot(z'_k-z'_i)^{\mathrm{T}}(z'_k-z'_i)},$$

这里我们根据三角形的某些边的点积来表示 $\cos\theta_i$. 若假定 δz_i 与 z_i 相比是小的，则可线性化这个约束如下：通过用分式的分母相乘，乘遍所有项得到一个所有的“δ-变量”(如 δx_i)的四次多项式，并且丢掉所有包括多于一个 δ-变量作为因子的项，这就得到其中所有的 δ-变量线性出现的方程. 如果把所有这些根据新的角度和距离测量的线性约束集中在一起，我们得到一个所有 δ-变量的超定线性方程组. 希望求出最小的校正，即几乎差不多都满足些约束的最小的 δx_i 值等等. 这是一个最小二乘问题. ◇

104

1. 在统计学中的标准记号不同于线性代数，统计学家记作 $X\boldsymbol{\beta}=\boldsymbol{y}$ 而不是 $A\boldsymbol{x}=\boldsymbol{b}$.

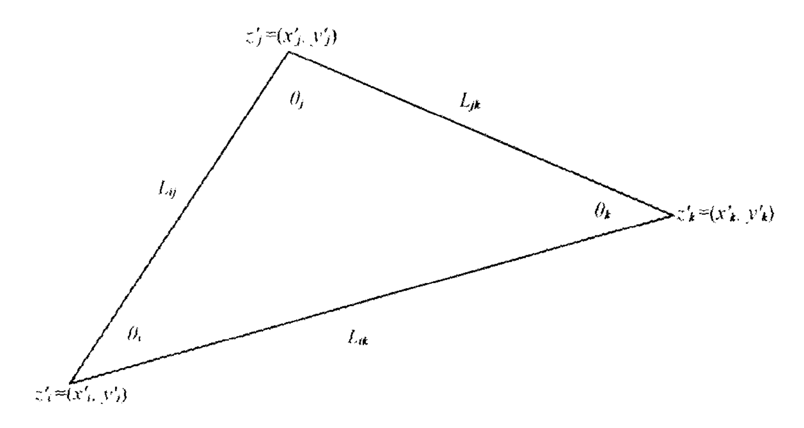


图 3-2 更新测量数据库中的约束

稍后，我们在引进更多的方法以后，也将指出怎样把图像压缩(image compression)解释为一个最小二乘问题(见例 3.4).

3.2 解线性最小二乘问题的矩阵分解

线性最小二乘问题有下面几种显式解法：

1. 正规方程；
2. QR 分解；
3. SVD；
4. 变换到一个线性方程组(见问题 3.3).

第一种方法最快但最不准确；当条件数小时，它比较适当. 第二种方法是一种标准的方法，其代价等于第一种方法的两倍. 第三种方法对病态问题即当 $\boldsymbol{A}$ 不是满秩时具有最大的使用价值，它多花费好几倍代价. 当问题病态时，最后的方法让我们做迭代精化去改善解. 除第三种外所有的方法都可适用于有效地处理稀疏矩阵[33]. 我们将依次讨论每种解法. 对方法 1 和 2 开始假定 $\boldsymbol{A}$ 有列满秩 n. 105

3.2.1 正规方程

为导出正规方程，我们寻找使梯度 $\|\boldsymbol{Ax}-\boldsymbol{b}\|_2^2=(\boldsymbol{Ax}-\boldsymbol{b})^{\mathrm{T}}(\boldsymbol{Ax}-\boldsymbol{b})$ 为零的 $\boldsymbol{x}$. 故要求

$$0=\lim_{e\to 0}\frac{(\boldsymbol{A}(\boldsymbol{x}+\boldsymbol{e})-\boldsymbol{b})^{\mathrm{T}}(\boldsymbol{A}(\boldsymbol{x}+\boldsymbol{e})-\boldsymbol{b})-(\boldsymbol{Ax}-\boldsymbol{b})^{\mathrm{T}}(\boldsymbol{Ax}-\boldsymbol{b})}{\|\boldsymbol{e}\|_2}$$

$$=\lim_{e\to 0}\frac{2\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Ax}-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b})+\boldsymbol{e}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Ae}}{\|\boldsymbol{e}\|_2}.$$

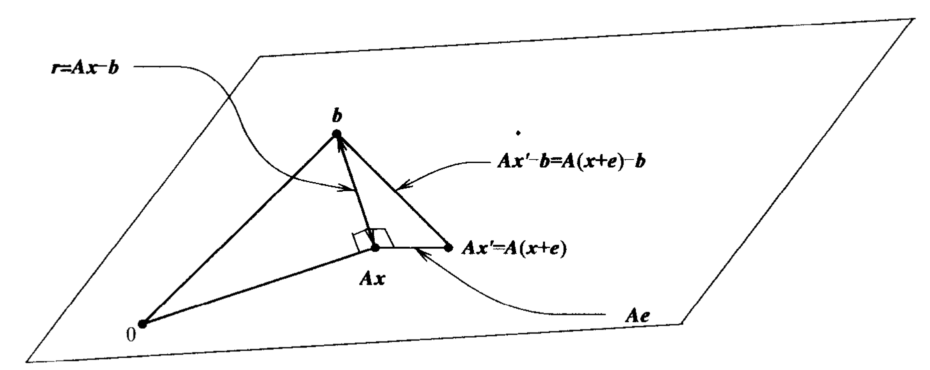
当 $\boldsymbol{e}$ 趋于 0 时第二项 $\dfrac{|\boldsymbol{e}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Ae}|}{\|\boldsymbol{e}\|_2}\leqslant\dfrac{\|\boldsymbol{A}\|_2^2\|\boldsymbol{e}\|_2^2}{\|\boldsymbol{e}\|_2}=\|\boldsymbol{A}\|_2^2\|\boldsymbol{e}\|_2$ 接近于 0，故第一项中因

子 A^TAx-A^Tb 也必须为0，或者 $A^TAx=A^Tb$. 这就是一个具有 n 个未知量的 n 个线性方程的方程组——正规方程.

为什么 $x=(A^TA)^{-1}A^Tb$ 是使 $\|Ax-b\|_2^2$ 达到极小的向量？可以注意黑塞(Hessian)矩阵 A^TA 是正定的，这意味着函数是严格凸的并且任何临界点是整体极小的. 或者记 $x'=x+e$ 即可完成平方并简化

$$\begin{aligned}(Ax'-b)^T(Ax'-b)&=(Ae+Ax-b)^T(Ae+Ax-b)\\&=(Ae)^T(Ae)+(Ax-b)^T(Ax-b)+2(Ae)^T(Ax-b)\\&=\|Ae\|_2^2+\|Ax-b\|_2^2+2e^T(A^TAx-A^Tb)\\&=\|Ae\|_2^2+\|Ax-b\|_2^2.\end{aligned}$$

显然当 $e=0$ 时上式达到极小. 这正好是毕达哥拉斯定理，正如下面说明的那样因为残差 $r=Ax-b$ 正交于 A 的列张成的空间，即 $0=A^Tr=A^TAx-A^Tb$ [所示的平面是 A 的列向量张成的以致 Ax，Ae 和 $Ax'=A(x+e)$ 全部位于平面内]



106

因为 A^TA 对称正定，所以可利用楚列斯基分解求解正规方程. 计算 A^TA，A^Tb 和楚列斯基分解的总代价是 $n^2m+\frac{1}{3}n^3+O(n^2)$ flops. 因为 $m\geq n$，所以构成 A^TA 的代价 n^2m 占主要地位.

3.2.2 QR分解

定理3.1 QR分解. 设 A 是 $m\times n$ 阶矩阵，$m\geq n$. 假如 A 为列满秩的，则存在一个唯一的正交阵 $Q(Q^TQ=I_n)$ 和唯一的具有正对角元 $r_{ii}>0$ 的 $n\times n$ 阶上三角阵 R 使得 $A=QR$.

证明 我们给出这个定理的两个证明. 第一，本定理是格拉姆-施密特正交化方法的一个重述[139]. 若对 $A=[a_1,a_2,\cdots,a_n]$ 的列 a_i 从左到右应用格拉姆-施密特正交化过程，我们得到张成同一个空间的一系列正交向量 q_1 到 q_n：这些正交向量是 Q 的列. 格拉姆-施密特过程也计算系数 $r_{ji}=q_j^Ta_i$，把每列的 a_i 表达为 q_1 到 q_i 的一个线性组合：$a_i=\sum_{j=1}^{i}r_{ji}q_j$. r_{ji} 正好是 R 的元素.

算法 3.1 分解 $A=QR$ 的经典格拉姆-施密特算法(CGS)和修正的格拉姆-施密特算法(MGS):

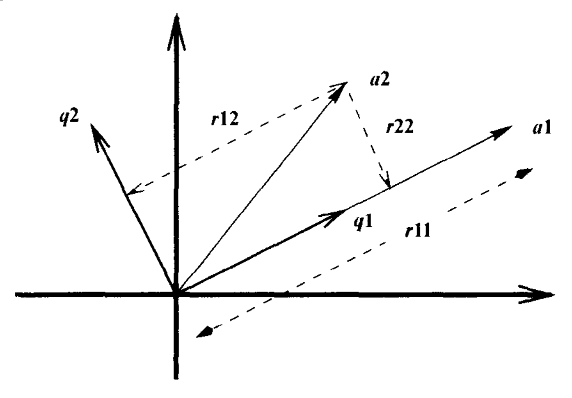
for $i=1$ to n/* 计算 Q 和 R 的第 i 列 */
  $q_i=a_i$
  for $j=1$ to $i-1$/* 从 a_i 中减去 q_j 方向中的分量 */
    $\begin{cases} r_{ji}=q_j^{\mathrm{T}}a_i & \text{CGS} \\ r_{ji}=q_j^{\mathrm{T}}q_i & \text{MGS} \end{cases}$
    $q_i=q_i-r_{ji}q_j$
  end for
  $r_{ii}=\|q_i\|_2$
  if $r_{ii}=0$/* a_i 与 $a_1, \cdots, a_{i-1}$ 线性相关 */
    quit
  end if
  $q_i=q_i/r_{ii}$
end for

算法中两个计算 r_{ji} 的公式在数学上是等价的，我们把它留作练习(见问题 3.1). 若 A 为列满秩的，则 r_{ii} 将不为 0. 下图说明当 A 是 2×2 阶矩阵时的格拉姆-施密特正交化过程.

107

本定理的第 2 个证明将利用算法 3.2，它在 3.4.1 节中提出. □

遗憾的是，当 A 的列几乎线性相关时 CGS 按浮点算术运算数值是不稳定的. MGS 是较为稳定的，将被用于本书后面的算法中，但是当 A 为病态阵时在 Q 中的结果仍然可能偏离正交($\|Q^{\mathrm{T}}Q-I\|$远大于 ε)[31,32,33,149]. 3.4.1 节中的算法 3.2 是另一个分解 $A=QR$ 的算法，见问题 3.2.



我们将利用三种稍有不同的方式分解 $A=QR$ 推导使 $\|Ax-b\|_2$ 达到极小的 x 的公式. 首先，我们总可选择另外的 $m-n$ 个标准正交向量 $\tilde{Q}$ 使得 $[Q,\tilde{Q}]$ 是一个正交方阵(例如，可选择我们需要的另外的 $m-n$ 个无关的向量 $\tilde{X}$，然后应用算法3.1于 $n\times n$ 阶非奇异阵 $[Q,\tilde{X}]$). 于是

$$
\begin{aligned}
\|Ax-b\|_2^2 &= \|[Q,\tilde{Q}]^{\mathrm{T}}(Ax-b)\|_2^2 \quad \text{由引理 1.7 的第 4 部分} \\
&= \left\| \begin{bmatrix} Q^{\mathrm{T}} \\ \tilde{Q}^{\mathrm{T}} \end{bmatrix} (QRx-b) \right\|_2^2 \\
&= \left\| \begin{bmatrix} I^{n\times n} \\ O^{(m-n)\times n} \end{bmatrix} Rx - \begin{bmatrix} Q^{\mathrm{T}}b \\ \tilde{Q}^{\mathrm{T}}b \end{bmatrix} \right\|_2^2 \\
&= \left\| \begin{bmatrix} Rx-Q^{\mathrm{T}}b \\ -\tilde{Q}^{\mathrm{T}}b \end{bmatrix} \right\|_2^2 \\
&= \|Rx-Q^{\mathrm{T}}b\|_2^2 + \|\tilde{Q}^{\mathrm{T}}b\|_2^2 \\
&\geqslant \|\tilde{Q}^{\mathrm{T}}b\|_2^2.
\end{aligned}
$$

可以解 $Rx-Q^{\mathrm{T}}b=0$ 得到 x，因为 A 和 R 有相同的秩 n，故 R 非奇异. 因而 $x=R^{-1}Q^{\mathrm{T}}b$, $\|Ax-b\|_2$ 的极小值是 $\|\tilde{Q}^{\mathrm{T}}b\|_2$.

[108] 下面是不利用矩阵 $\tilde{Q}$ 的第2种稍为不同的推导. 改写 $Ax-b$ 为

$$
\begin{aligned}
Ax-b &= QRx-b = QRx-(QQ^{\mathrm{T}}+I-QQ^{\mathrm{T}})b \\
&= Q(Rx-Q^{\mathrm{T}}b)-(I-QQ^{\mathrm{T}})b.
\end{aligned}
$$

注意向量 $Q(Rx-Q^{\mathrm{T}}b)$ 和 $(I-QQ^{\mathrm{T}})b$ 是正交的，因为 $(Q(Rx-Q^{\mathrm{T}}b))^{\mathrm{T}}((I-QQ^{\mathrm{T}})b)=(Rx-Q^{\mathrm{T}}b)^{\mathrm{T}}[Q^{\mathrm{T}}(I-QQ^{\mathrm{T}})]b=(Rx-Q^{\mathrm{T}}b)^{\mathrm{T}}[0]b=0$. 所以，由毕达哥拉斯定理，

$$
\begin{aligned}
\|Ax-b\|_2^2 &= \|Q(Rx-Q^{\mathrm{T}}b)\|_2^2 + \|(I-QQ^{\mathrm{T}})b\|_2^2 \\
&= \|Rx-Q^{\mathrm{T}}b\|_2^2 + \|(I-QQ^{\mathrm{T}})b\|_2^2.
\end{aligned}
$$

其中我们利用了引理1.7的第4部分中的形式 $\|Qy\|_2^2=\|y\|_2^2$. 当第1项为零即 $x=R^{-1}Q^{\mathrm{T}}b$ 时，这个平方和达到极小.

最后是从正规方程解出发的第3种推导：

$$
\begin{aligned}
x &= (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b \\
&= (R^{\mathrm{T}}Q^{\mathrm{T}}QR)^{-1}R^{\mathrm{T}}Q^{\mathrm{T}}b = (R^{\mathrm{T}}R)^{-1}R^{\mathrm{T}}Q^{\mathrm{T}}b \\
&= R^{-1}R^{-\mathrm{T}}R^{\mathrm{T}}Q^{\mathrm{T}}b = R^{-1}Q^{\mathrm{T}}b.
\end{aligned}
$$

稍后将指出这个分解以及随后的最小二乘法解的代价是 $2n^2m-\frac{2}{3}n^3$，若 $m\gg n$ 则大约二倍于正规方程的代价. 若 $m=n$，则代价大约相同.

3.2.3 奇异值分解

SVD 是极其重要的分解，它被用于除求解最小二乘问题之外的许多方面.

定理 3.2 SVD. 设 A 是一个任意的 $m\times n$ 阶矩阵，$m\geq n$. 则可记 $A=U\Sigma V^{\mathrm{T}}$，其中 U 是 $m\times n$ 阶矩阵且满足 $U^{\mathrm{T}}U=I$，V 是 $n\times n$ 阶矩阵且满足 $V^{\mathrm{T}}V=I$，$\Sigma=\operatorname{diag}(\sigma_1,\cdots,\sigma_n)$，其中 $\sigma_1\geq\cdots\geq\sigma_n\geq 0$. U 的列 $u_1,\cdots,u_n$ 称为左奇异向量. V 的列 $v_1,\cdots,v_n$ 称为右奇异向量. σ_i 称为奇异值. (若 $m<n$，则考虑 A^{T} 定义的 SVD).

本定理几何上的重新表述如下：给定任意 $m\times n$ 阶矩阵 A，将它看作向量 $x\in\mathbb{R}^n$ 到向量 $y=Ax\in\mathbb{R}^m$ 的映射. 然后可以选择 $\mathbb{R}^n$ 的一个正交坐标系(其中单位轴是 V 的列)和另一个 $\mathbb{R}^m$ 的正交坐标系(其中单位轴是 U 的列)使得 A 是对角阵(Σ)，即把向量 $x=\sum_{i=1}^{n}\beta_i v_i$ 映射到 $y=Ax=\sum_{i=1}^{n}\sigma_i\beta_i u_i$. 换言之，倘若我们对矩阵的定义域和值域取适当的正交坐标系，则任何矩阵都是对角的.

109

证明 对 m 和 n 用归纳法：假定对 $(m-1)\times(n-1)$ 阶矩阵 SVD 存在，我们对 $m\times n$ 阶矩阵证明它存在. 假定 $A\neq 0$；否则可取 $\Sigma=0$ 并设 U 和 V 是任意的正交阵.

当 $n=1$ 时进行基础步骤(因为 $m\geq n$). 我们记 $A=U\Sigma V^{\mathrm{T}}$，其中 $U=A/\|A\|_2$，$\Sigma=\|A\|_2$，$V=1$.

对归纳步骤，选择 v 使 $\|v\|_2=1$ 和 $\|A\|_2=\|Av\|_2>0$. 由定义 $\|A\|_2=\max_{\|v\|_2=1}\|Av\|_2$. 这样的一个 v 存在. 设 $u=\dfrac{Av}{\|Av\|_2}$，这是一个单位向量. 选择 $\widetilde{U}$ 和 $\widetilde{V}$ 使得 $U=[u,\widetilde{U}]$ 是一个 $m\times m$ 阶正交阵，以及 $V=[v,\widetilde{V}]$ 是一个 $n\times n$ 阶正交阵. 今记

$$U^{\mathrm{T}}AV=\begin{bmatrix}u^{\mathrm{T}}\\ \widetilde{U}^{\mathrm{T}}\end{bmatrix}\cdot A\cdot[v\quad \widetilde{V}]=\begin{bmatrix}u^{\mathrm{T}}Av & u^{\mathrm{T}}A\widetilde{V}\\ \widetilde{U}^{\mathrm{T}}Av & \widetilde{U}^{\mathrm{T}}A\widetilde{V}\end{bmatrix}$$

于是

$$u^{\mathrm{T}}Av=\frac{(Av)^{\mathrm{T}}(Av)}{\|Av\|_2}=\frac{\|Av\|_2^2}{\|Av\|_2}=\|Av\|_2=\|A\|_2\equiv\sigma$$

且 $\widetilde{U}^{\mathrm{T}}Av=\widetilde{U}^{\mathrm{T}}u\|Av\|_2=0$. 我们也断定 $u^{\mathrm{T}}A\widetilde{V}=0$，因为否则 $\sigma=\|A\|_2=\|U^{\mathrm{T}}AV\|_2\geq\|[1,0,\cdots,0]U^{\mathrm{T}}AV\|_2=\|[\sigma|u^{\mathrm{T}}A\widetilde{V}]\|_2>\sigma$ 矛盾. (使用了引理 1.7 的第 7 部分.)

故 $U^{\mathrm{T}}AV=\begin{bmatrix}\sigma & 0\\ 0 & \widetilde{U}^{\mathrm{T}}A\widetilde{V}\end{bmatrix}=\begin{bmatrix}\sigma & 0\\ 0 & \widetilde{A}\end{bmatrix}$. 现在可应用对 $\widetilde{A}$ 的归纳假设得到 $\widetilde{A}=U_1\Sigma_1V_1^{\mathrm{T}}$，其中 U_1 是 $(m-1)\times(m-1)$ 阶矩阵，Σ_1 是 $(n-1)\times(n-1)$ 阶矩阵而 V_1 是 $(n-1)\times(n-1)$ 阶矩阵. 故

$$U^{\mathrm{T}}AV=\begin{bmatrix}\sigma & 0\\ 0 & U_1\Sigma_1 V_1^{\mathrm{T}}\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & U_1\end{bmatrix}\begin{bmatrix}\sigma & 0\\ 0 & \Sigma_1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & V_1\end{bmatrix}^{\mathrm{T}}$$

或

$$A=\left(U\begin{bmatrix}1 & 0\\ 0 & U_1\end{bmatrix}\right)\begin{bmatrix}\sigma & 0\\ 0 & \Sigma_1\end{bmatrix}\left(V\begin{bmatrix}1 & 0\\ 0 & V_1\end{bmatrix}\right)^{\mathrm{T}},$$

这就是我们所要求的分解. □

SVD 有大量重要的代数和几何性质，我们在此陈述其中最重要的性质.

定理 3.3 设 $A=U\Sigma V^{\mathrm{T}}$ 是 $m\times n$ 阶矩阵 A 的 SVD，其中 $m\geq n$.（对 $m<n$ 有类似的结果.）

1. 假如 A 对称，具有特征值 λ_i 和标准正交特征向量 u_i. 换言之 $A=U\Lambda U^{\mathrm{T}}$ 是 A 的一个特征分解，其中 $\Lambda=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)$，$U=[u_1,\cdots,u_n]$ 和 $UU^{\mathrm{T}}=I$. 则 A 的一个 SVD 是 $A=U\Sigma V^{\mathrm{T}}$，其中 $\sigma_i=|\lambda_i|$，$v_i=\mathrm{sign}(\lambda_i)u_i$，$\mathrm{sign}(0)=1$.

[110]

2. 对称阵 $A^{\mathrm{T}}A$ 的特征值是 σ_i^2. 右奇异向量 v_i 是对应的标准正交特征向量.

3. 对称阵 AA^{T} 的特征值是 σ_i^2 和 $m-n$ 个零. 左奇异向量 u_i 是特征值 σ_i^2 对应的标准正交向量. 对特征值 0 可取任意 $m-n$ 个正交向量作为特征向量.

4. 设 $H=\begin{bmatrix}0 & A^{\mathrm{T}}\\ A & 0\end{bmatrix}$，其中 A 是方阵且 $A=U\Sigma V^{\mathrm{T}}$ 是 A 的 SVD. 设 $\Sigma=\mathrm{diag}(\sigma_1,\cdots,\sigma_n)$，$U=[u_1,\cdots,u_n]$ 和 $V=[v_1,\cdots,v_n]$. 则 H 的 $2n$ 个特征值是 $\pm\sigma_i$，具有相应的单位特征向量 $\frac{1}{\sqrt{2}}\begin{bmatrix}v_i\\ \pm u_i\end{bmatrix}$.

5. 若 A 满秩，则 $\min_x\|Ax-b\|_2$ 的解是 $x=V\Sigma^{-1}U^{\mathrm{T}}b$.

6. $\|A\|_2=\sigma_1$. 若 A 是非奇异方阵，则 $\|A^{-1}\|_2^{-1}=\sigma_n$ 且 $\|A\|_2\cdot\|A^{-1}\|_2=\frac{\sigma_1}{\sigma_n}$.

7. 假定 $\sigma_1\geq\cdots\geq\sigma_r>\sigma_{r+1}=\cdots=\sigma_n=0$. 则 A 的秩为 r. A 的零空间即使 $Av=0$ 的向量 v 的子空间是由 V 的第 $r+1$ 列到第 n 列生成的空间：$\mathrm{span}(v_{r+1},\cdots,v_n)$. A 的值域，即对一切 w，形如 Aw 的向量的子空间是由 U 的第 1 列到第 r 列生成的空间：$\mathrm{span}(u_1,\cdots,u_r)$.

8. 设 S^{n-1} 是 $\mathbb{R}^n$ 中单位球面：$S^{n-1}=\{x\in\mathbb{R}^n:\|x\|_2=1\}$. 设 $A\cdot S^{n-1}$ 是在 A 之下 S^{n-1} 的像：$A\cdot S^{n-1}=\{Ax:x\in\mathbb{R}^n 且 \|x\|_2=1\}$. 则 $A\cdot S^{n-1}$ 是一个中心在 $\mathbb{R}^m$ 的原点，具有主轴 $\sigma_i u_i$ 的椭球.

9. 记 $V=[v_1,v_2,\cdots,v_n]$ 和 $U=[u_1,u_2,\cdots,u_n]$，故 $A=U\Sigma V^{\mathrm{T}}=\sum_{i=1}^{n}\sigma_i u_i v_i^{\mathrm{T}}$（秩 1 矩阵之和）. 还有最接近于 A（用 $\|\cdot\|_2$ 度量）且秩为 $k<n$ 的一个矩阵是 $A_k=$

$\sum_{i=1}^{k}\sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathrm{T}}$，且$\|A-A_k\|_2=\sigma_{k+1}$. 也可记作 $A_k=U\Sigma_k V^{\mathrm{T}}$，其中 $\Sigma_k=\operatorname{diag}(\sigma_1,\cdots,\sigma_k,0,\cdots,0)$.

证明

(1) 由 SVD 定义可知成立.

(2) $A^{\mathrm{T}}A=V\Sigma U^{\mathrm{T}}U\Sigma V^{\mathrm{T}}=V\Sigma^2V^{\mathrm{T}}$. 这是 $A^{\mathrm{T}}A$ 的一个特征分解，用 V 的列为特征向量并利用Σ^2的对角元为特征值.

(3) 选择 $m\times(m-n)$阶矩阵 $\widetilde{U}$ 使得$[U,\widetilde{U}]$是方的且为正交的. 然后记

$$AA^{\mathrm{T}}=U\Sigma V^{\mathrm{T}}V\Sigma U^{\mathrm{T}}=U\Sigma^2U^{\mathrm{T}}=[U,\widetilde{U}]\begin{bmatrix}\Sigma^2 & 0\\ 0 & 0\end{bmatrix}[U,\widetilde{U}]^{\mathrm{T}}.$$

111

这就是 AA^{T} 的一个特征分解.

(4) 见问题3.14.

(5) $\|Ax-b\|_2^2=\|U\Sigma V^{\mathrm{T}}x-b\|_2^2$. 因为 A 满秩，故Σ也满秩，从而Σ可逆. 如上所设$[U,\widetilde{U}]$是方的且正交的，故

$$\begin{aligned}\|U\Sigma V^{\mathrm{T}}x-b\|_2^2&=\left\|\begin{bmatrix}U^{\mathrm{T}}\\ \widetilde{U}^{\mathrm{T}}\end{bmatrix}(U\Sigma V^{\mathrm{T}}x-b)\right\|_2^2\\&=\left\|\begin{bmatrix}\Sigma V^{\mathrm{T}}x-U^{\mathrm{T}}b\\ -\widetilde{U}^{\mathrm{T}}b\end{bmatrix}\right\|_2^2\\&=\|\Sigma V^{\mathrm{T}}x-U^{\mathrm{T}}b\|_2^2+\|\widetilde{U}^{\mathrm{T}}b\|_2^2.\end{aligned}$$

取第一项为零，即 $x=V\Sigma^{-1}U^{\mathrm{T}}b$，上式达到极小.

(6) 由对角阵的2-范数是其对角线上绝对值最大的元素这个定义知，这是显然的. 因而，由引理1.7第3部分得

$$\|A\|_2=\|U^{\mathrm{T}}AV\|_2=\|\Sigma\|_2=\sigma_1,\|A^{-1}\|_2=\|V^{\mathrm{T}}A^{-1}U\|_2=\|\Sigma^{-1}\|_2=\sigma_n^{-1}$$

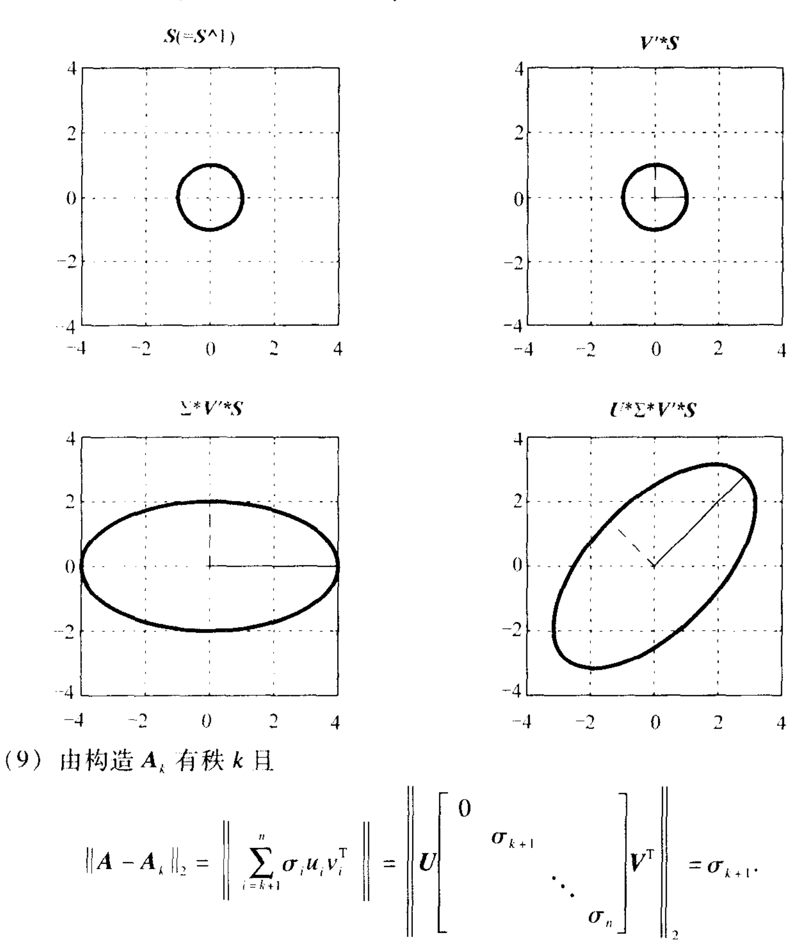
(7) 再次选择一个 $m\times(m-n)$阶矩阵 $\widetilde{U}$，使得 $m\times m$ 阶矩阵 $\hat{U}=[U,\widetilde{U}]$是正交的. 因为 $\hat{U}$ 和 V 非奇异，所以 A 和 $\hat{U}^{\mathrm{T}}AV=\begin{bmatrix}\Sigma^{n\times n}\\ 0^{(m-n)\times n}\end{bmatrix}\equiv\hat{\Sigma}$有相同的秩，由$\Sigma$的假设知其秩为 r. 还有因为 $Av=0$ 当且仅当 $\hat{U}^{\mathrm{T}}AV(V^{\mathrm{T}}v)=0$，所以 v 是在 A 的零空间中当且仅当 $V^{\mathrm{T}}v$ 是在 $\hat{U}^{\mathrm{T}}AV=\hat{\Sigma}$的零空间中. 但是$\hat{\Sigma}$的零空间显然是由 $n\times n$ 阶单位阵 I_n 的第 $r+1$ 列到第 n 列所张成的，故 A 的零空间是由 V 乘这些列即 v_{r+1}到 v_n 所张成的. 类似的论证表明 A 的值域和 $\hat{U}$ 乘 $\hat{U}^{\mathrm{T}}AV=\hat{\Sigma}$的值域相同，即 $\hat{U}$ 乘 I_m 的前 r 列或 u_1 到 u_r.

(8) 通过一次用 $A=U\Sigma V^{\mathrm{T}}$ 的一个因子相乘来“构造”集 $A\cdot S^{n-1}$. 当

$$A=\begin{bmatrix}3&1\\1&3\end{bmatrix}=\begin{bmatrix}2^{-1/2}&-2^{-1/2}\\2^{-1/2}&2^{-1/2}\end{bmatrix}\cdot\begin{bmatrix}4&0\\0&2\end{bmatrix}\cdot\begin{bmatrix}2^{-1/2}&-2^{-1/2}\\2^{-1/2}&2^{-1/2}\end{bmatrix}^{\mathrm{T}}\equiv U\Sigma V^{\mathrm{T}}.$$

时，下面的图形说明所出现的情况．为简单起见假定 A 是非奇异方阵．因为 V 是正交的，故映射单位向量为另一个单位向量．$V^{\mathrm{T}}\cdot S^{n-1}=S^{n-1}$．其次，因为 $v\in S^{n-1}$ 当且仅当 $\|v\|_2=1$，$w\in\Sigma S^{n-1}$ 当且仅当 $\|\Sigma^{-1}w\|_2=1$ 或 $\sum_{i=1}^n(w_i/\sigma_i)^2=1$．这就确定了一个具有主轴 σ_ie_i 的椭球，其中 e_i 是单位阵的第 i 列．最后，用 U 乘每个 $w=\Sigma v$ 正好旋转椭圆使得每个 e_i 变成 U 的第 i 列 u_i.

112



（9）由构造 A_k 有秩 k 且

$$\|A-A_k\|_2=\left\|\sum_{i=k+1}^{n}\sigma_iu_iv_i^{\mathrm{T}}\right\|=\left\|U\begin{bmatrix}0&&&\\&\sigma_{k+1}&&\\&&\ddots&\\&&&\sigma_n\end{bmatrix}V^{\mathrm{T}}\right\|_2=\sigma_{k+1}.$$

余下的是证明不存在更接近于 A 的秩为 k 的矩阵．设 B 是任意的秩为 k 的矩阵，故它的零空间维数为 $n-k$．由 $\{v_1,\cdots,v_{k+1}\}$ 张成的空间维数为 $k+1$．因为这两个空间维数之和为 $(n-k)+(k+1)>n$，所以这两个空间必定重叠．设 h 是它们的交集中的一个单位向量．则

$$\|A-B\|_2^2 \geqslant \|(A-B)h\|_2^2 = \|Ah\|_2^2 = \|U\Sigma V^{\mathrm{T}}h\|_2^2$$
$$= \|\Sigma(V^{\mathrm{T}}h)\|_2^2 \geqslant \sigma_{k+1}^2 \|V^{\mathrm{T}}h\|_2^2 = \sigma_{k+1}^2. \qquad \square$$

例 3.4 我们利用图像压缩(image compression)来说明定理 3.3 的最后部分. 特别地，我们将用一名小丑图像的低秩逼近来说明它. 一个 $m\times n$ 图像正好是一个 $m\times n$ 阶矩阵，其中元素 (i,j) 被解释为像素 (i,j) 的亮度. 换言之，矩阵元素(譬如说)从 0 变到 1 被解释成像素从黑色(=0)经过各种浓淡的灰色变到白色(=1)(彩色也是可能的). 我们通常不去存放或传输图像的所有 $m\cdot n$ 个矩阵元素，而是存放极少的数去压缩图像，由这些数仍旧能近似地重构原来的图像. 正如我们现在说明的那样，可以利用定理 3.3 的第 9 部分去做这件事. [113]

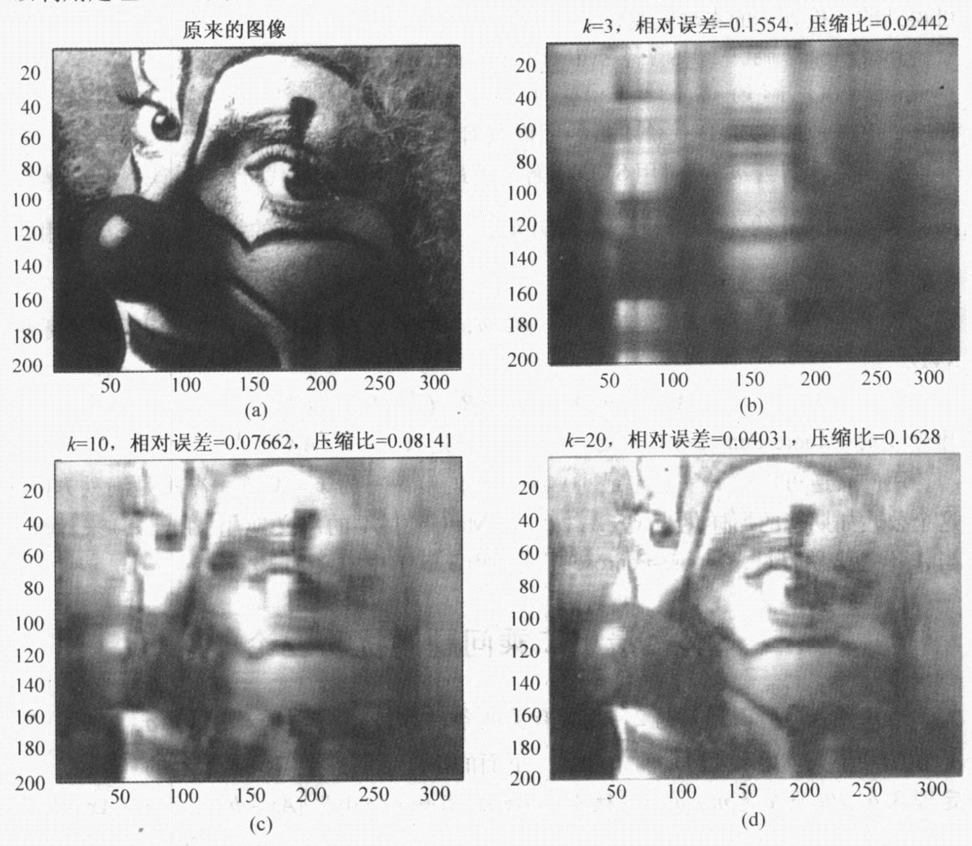


图 3-3 利用 SVD 的图像压缩. (a) 原来的图像. (b) Rank $k=3$ 近似. (c) Rank $k=10$ 近似. (d) Rank $k=20$ 近似

考察图 3-3a 中的图像. 这是对应于 320×200 阶矩阵 A 的 320×200 像素图像. 设 $A=U\Sigma V^{\mathrm{T}}$ 是 A 的 SVD. 定理 3.3 的第 9 部分告诉我们在极小化 $\|A-A_k\|_2=\sigma_{k+1}$ 意义下 $A_k=\sum_{i=1}^{k}\sigma_i u_i v_i^{\mathrm{T}}$ 是 A 的最佳秩 k 近似. 注意它只取 $m\cdot k+n\cdot k=(m+n)\cdot k$

个字存放 u_1 到 u_k 和 $\sigma_1 v_1$ 到 $\sigma_k v_k$，由这些字可以重构 A_k. 相反，它取 $m \cdot n$ 个字存放 A(或明确地存放 A_k)，当 k 小时，这个值相当大. 故我们将使用 A_k 和用 $(m+n) \cdot k$ 个字存放压缩图像. 在图3-3中其他的图像指出各种各样 k 值的近似及相对误差 σ_{k+1}/σ_1 和压缩比 $(m+n) \cdot k/(m \cdot n) = 520 \cdot k/64\,000 \approx k/123$.

k	相对误差 $= \sigma_{k+1}/\sigma_k$	压缩比 $= 520k/64\,000$
3	0.155	0.024
10	0.077	0.081
20	0.040	0.163

这些图像是利用下列指令产生的(小丑及其他图像在Matlab可视化示范文件中是可得到的. 检查你的本地安装位置)：

```
load clown. mat; [U,S,V] = svd(X); colormap('gray');
,image(U(: ,1 : k) * S(1 : k,1 : k) * V(: ,1 : k)')
```

还有许多其他的比SVD代价更低的图像压缩方法可以利用[189,152]. ◇

后面将看到当 $m \gg n$ 时用SVD求解一个最小二乘问题的代价大约与用QR分解相同，对较小的 m 约为 $4n^2m - \frac{4}{3}n^3 + O(n^2)$. QR和SVD代价的精确地比较与所用的机器有关，细节见3.6节.

定义3.1 假如 A 为 $m \times n$ 阶满秩阵，$m \geq n$，$A = QR = U\Sigma V^T$ 分别是 A 的QR分解和SVD. 则

114
~
116

$$A^+ \equiv (A^TA)^{-1}A^T = R^{-1}Q^T = V\Sigma^{-1}U^T$$

称为 A 的(Moore-Penrose)广义逆. 若 $m < n$，则 $A^+ \equiv A^T(AA^T)^{-1}$

用广义逆可以把满秩超定最小二乘问题的解简写为 $x = A^+b$. 若 A 是满秩方阵，这个公式如期待的那样化为 $x = A^{-1}b$. 在Matlab中 A 的广义逆用 pinv(A)来计算. 当 A 不是满秩阵时，Moore-Penrose广义逆由3.5节中定义3.2给出.

3.3 最小二乘问题的扰动理论

当 A 不是方阵时，其与2-范数有关的条件数是 $\kappa_2(A) \equiv \sigma_{\max}(A)/\sigma_{\min}(A)$. 当 A 是方阵时，这就化为通常的条件数. 下面的定理证明这个定义是恰当的.

定理3.4 假如 A 是 $m \times n$ 阶满秩阵，$m \geq n$. 假如 x 极小化 $\|Ax-b\|_2$，设 $r = Ax - b$ 是残差. $\tilde{x}$ 极小化 $\|(A+\delta A)\tilde{x} - (b+\delta b)\|_2$. 假定 $\varepsilon \equiv \max\left(\frac{\|\delta A\|_2}{\|A\|_2}, \frac{\|\delta b\|_2}{\|b\|_2}\right) < \frac{1}{\kappa_2(A)} = \frac{\sigma_{\min}(A)}{\sigma_{\max}(A)}$. 则

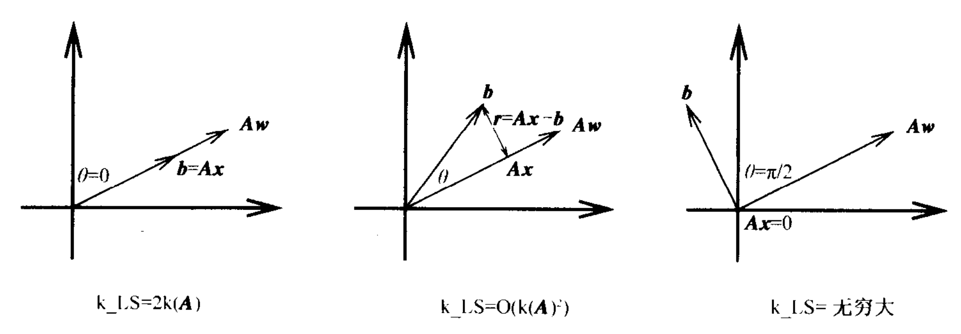
$$\frac{\|\tilde{x} - x\|_2}{\|x\|_2} \leq \varepsilon \cdot \left\{\frac{2 \cdot \kappa_2(A)}{\cos\theta} + \tan\theta \cdot \kappa_2^2(A)\right\} + O(\varepsilon^2) \equiv \varepsilon \cdot \kappa_{LS} + O(\varepsilon^2),$$

其中 $\sin\theta=\frac{\|r\|_2}{\|b\|_2}$. 换言之，$\theta$ 是向量 $\boldsymbol{b}$ 和 $\boldsymbol{Ax}$ 之间的夹角并且度量残差范数 $\|\boldsymbol{r}\|_2$ 是大（接近于 $\|\boldsymbol{b}\|$）还是小（接近于0）. κ_{LS} 是最小二乘问题的条件数.

证明的梗概 按 δA 和 δb 的幂展开 $\tilde{x}=((A+\delta A)^{\mathrm{T}}(A+\delta A))^{-1}(A+\delta A)^{\mathrm{T}}(b+\delta b)$，并舍弃除了 δA 和 δb 线性项之外所有项. □

我们以推导方的线性方程组 $Ax=b$ 的扰动解的界(2.4)同样的理由假定 $\varepsilon\cdot\kappa_2(A)<1$：它保证 $A+\delta A$ 满秩使得 $\tilde{x}$ 是唯一确定的.

我们可如下说明这个界. 若 θ 为0或非常小，则残差是小的并且实际的条件数大约是 $2\kappa_2(A)$，很像通常的解线性方程组的条件数. 若 θ 不是小的但不接近于 $\pi/2$，残差是适度地大的，从而实际的条件数可能大得多：$\kappa_2^2(A)$. 若 θ 接近于 $\pi/2$，那么真解几乎为0，则即使 $\kappa_2(A)$ 是小的而实际的条件数也会变成无界的. 这三种情况被说明如下. 右边的图使之容易看出为什么当 $\theta=\frac{\pi}{2}$ 时条件数是无穷大：此时解 $x=0$，在 A 和 b 中几乎任何任意小的改变将得到一个"无穷"大相对变化的非零解 x. 117



定理3.4中消去 $O(\varepsilon^2)$ 项的界的另一种形式如下[258,149]［这里 $\tilde{r}$ 是扰动后的残差，$\tilde{r}=(A+\delta A)\tilde{x}-(b+\delta b)$］：

$$\frac{\|\tilde{x}-x\|_2}{\|x\|_2}\leqslant\frac{\varepsilon\kappa_2(A)}{1-\varepsilon\kappa_2(A)}\left(2+(\kappa_2(A)+1)\frac{\|r\|_2}{\|A\|_2\|x\|_2}\right),$$

$$\frac{\|\tilde{r}-r\|_2}{\|r\|_2}\leqslant(1+2\varepsilon\kappa_2(A)).$$

我们将看到正确地执行的QR分解和SVD两者都是数值稳定的. 即它们得到一个解 $\tilde{x}$ 可极小化 $\|(A+\delta A)\tilde{x}-(b+\delta b)\|_2$ 与

$$\max\left(\frac{\|\delta A\|}{\|A\|},\frac{\|\delta b\|}{\|b\|}\right)=O(\varepsilon).$$

可以把这个结果与上面的扰动界合起来，得到很像求解线性方程组那样的最小二乘问题解的误差界.

正规方程不能作为精确的方法. 因为它们涉及求解 $(A^{\mathrm{T}}A)x=A^{\mathrm{T}}b$，其精度依赖于条件数 $\kappa_2(A^{\mathrm{T}}A)=\kappa_2^2(A)$. 因而误差总是以 $\kappa_2^2(A)\varepsilon$ 而决非正好是以 $\kappa_2(A)\varepsilon$ 为界.

所以，我们预期正规方程可能失去基于 QR 分解和 SVD 方法那样精度的数字位数的两倍.

进一步求解正规方程不一定是稳定的；即对小的 δA 和 δb 计算解 $\tilde{x}$ 一般不极小化 $\|(A+\delta A)\tilde{x}-(b+\delta b)\|_2$. 尽管如此，当条件数小时，我们认为正规方程大约像 QR 分解或 SVD 那样精确. 因为正规方程是求解最小二乘问题最快的方法，所以当矩阵良态时它们是可选的方法.

在 3.5 节中将回到求解非常病态的最小二乘问题的问题

3.4 正交矩阵

正如 3.2.2 节中所述，当正交化的向量几乎线性相关时，格拉姆－施密特正交化（算法 3.1）不可能计算一个正交阵 Q，故不能用它稳定地计算 QR 分解.

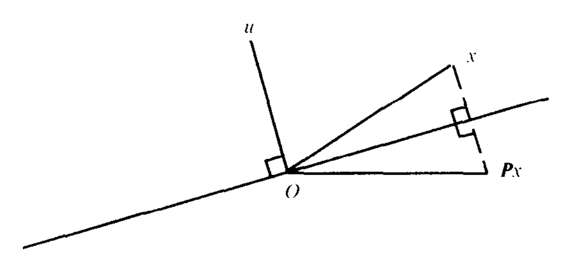
118

相反，基于某些称为豪斯霍尔德反射（Householder reflection）和吉文斯旋转（Givens rotation）的容易计算的正交阵的算法，可以选择零元素引入到相乘向量中. 后面将指出利用这些正交矩阵引进零元素的任何算法自动地是稳定的. 这个误差分析将应用于 QR 分解算法以及 SVD 算法和第 4 章和第 5 章中的特征值算法.

尽管 MGS 算法可能会产生非正交的 Q，但是，在数值线性代数中它还是有重要的用途.（CGS 因为欠稳定，故用途不大.）这些用途包括利用对分法和逆迭代（5.3.4 节）求对称三对角阵的特征向量，以及约化矩阵为某些“紧凑”形式的阿诺尔迪（Arnoldi）算法和兰乔斯算法（6.6.1、6.6.6 和 7.4 节）. Arnoldi 和兰乔斯算法被用于作为求解稀疏线性方程组和求稀疏矩阵特征值的算法的基础. MGS 也能被修正用来稳定地求解最小二乘问题，但 Q 仍旧可能偏离正交[33].

3.4.1 豪斯霍尔德变换

豪斯霍尔德变换（或反射）是一个形如 $P=I-2uu^{\mathrm{T}}$ 的矩阵，其中 $\|u\|_2=1$. 容易看出 $P=P^{\mathrm{T}}, PP^{\mathrm{T}}=(I-2uu^{\mathrm{T}})(I-2uu^{\mathrm{T}})=I-4uu^{\mathrm{T}}+4uu^{\mathrm{T}}uu^{\mathrm{T}}=I$，故 P 是一个对称正交矩阵. 因为 Px 是 x 在经过 O 点垂直于 u 的平面中的反射，所以 P 被称为反射.



给定一个向量 x，容易找到一个豪斯霍尔德反射 $P=I-2uu^{\mathrm{T}}$ 使得 x 除第一个元素外其余元素都为零：$Px=[c,0,\cdots,0]^{\mathrm{T}}=c\cdot e_1$. 做法如下. 记 $Px=x-2u(u^{\mathrm{T}}x)=$

$c \cdot e_1$，因此$u = \frac{1}{2(u^{\mathrm{T}} x)}(x - ce_1)$；即 u 是 x 和 e_1 的线性组合. 因为$\|x\|_2 = \|Px\|_2 = |c|$，所以 u 必须平行于向量 $\tilde{u} = x \pm \|x\|_2 e_1$，故 $u = \tilde{u}/\|\tilde{u}\|_2$. 我们可以验证只要 $\tilde{u} \neq 0$，选取任一符号得到满足 $Px = ce_1$ 的一个 u. 我们将使用 $\tilde{u} = x + \mathrm{sign}(x_1) e_1$，因为这意味着在计算 $\tilde{u}$ 的第一个分量中不存在对消. 总之，得到

119

$$\tilde{u} = \begin{bmatrix} x_1 + \mathrm{sign}(x_1) \cdot \|x\|_2 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad u = \frac{\tilde{u}}{\|\tilde{u}\|_2}.$$

把这个过程记为 $u = \mathrm{House}(x)$.（实际上，可以存放 $\tilde{u}$ 而不是 u，以节省计算 u 的工作量，并利用公式 $P = I - (2/\|\tilde{u}\|_2^2)\tilde{u}\tilde{u}^{\mathrm{T}}$ 而不是用 $P = I - 2uu^{\mathrm{T}}$.）

例 3.5 说明如何利用豪斯霍尔德变换计算 5×4 阶矩阵 A 的 QR 分解. 本例将对一般的 $m \times n$ 阶矩阵给出明显的模式. 在下面的矩阵中，P_i 是一个 5×5 正交阵. x 表示一般的非零元，o 表示零元.

1. 选择 P_1 然后
$$A_1 \equiv P_1 A = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & x & x & x \\ o & x & x & x \\ o & x & x & x \end{bmatrix}.$$

2. 选择 $P_2 = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & P'_2 \end{array}\right]$ 使
$$A_2 \equiv P_2 A_1 = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & x & x \end{bmatrix}.$$

3. 选择 $P_3 = \left[\begin{array}{cc|c} 1 & & 0 \\ & 1 & \\ \hline \multicolumn{2}{c|}{0} & P'_3 \end{array}\right]$ 使
$$A_3 \equiv P_3 A_2 = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & o & x \\ o & o & o & x \end{bmatrix}.$$

4. 选择 $P_4 = \left[\begin{array}{ccc|c} 1 & & & \\ & 1 & & 0 \\ & & 1 & \\ \hline \multicolumn{3}{c|}{0} & P'_4 \end{array}\right]$ 使
$$A_4 \equiv P_4 A_3 = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & o & x \\ o & o & o & o \end{bmatrix}.$$

这里，选择豪斯霍尔德阵 P'_i 使第 i 列次对角线下的元素为零，这样做不破坏前

面列中已经引进的零元素.

我们称最后的 5×4 阶上三角阵为 $\widetilde{\boldsymbol{R}}\equiv\boldsymbol{A}_4$. 而且 $\boldsymbol{A}=\boldsymbol{P}_1^{\mathrm{T}}\boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{P}_3^{\mathrm{T}}\boldsymbol{P}_4^{\mathrm{T}}\widetilde{\boldsymbol{R}}=\boldsymbol{QR}$, 其中 $\boldsymbol{Q}$ 是 $\boldsymbol{P}_1^{\mathrm{T}}\boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{P}_3^{\mathrm{T}}\boldsymbol{P}_4^{\mathrm{T}}=\boldsymbol{P}_1\boldsymbol{P}_2\boldsymbol{P}_3\boldsymbol{P}_4$(因为所有的 $\boldsymbol{P}_i$ 是对称的)的前四列而 $\boldsymbol{R}$ 是 $\widetilde{\boldsymbol{R}}$ 的前

120 四行. ◇

下面是利用豪斯霍尔德变换作 QR 分解的一般算法.

算法 3.2 利用豪斯霍尔德反射作 QR 分解:

for $i=1$ to $\min(m-1,n)$

$\quad u_i=\mathrm{House}(\boldsymbol{A}(i:m,i))$

$\quad \boldsymbol{P}'_i=\boldsymbol{I}-2\boldsymbol{u}_i\boldsymbol{u}_i^{\mathrm{T}}$

$\quad \boldsymbol{A}(i:m,i:n)=\boldsymbol{P}_i'\boldsymbol{A}(i:m,i:n)$

end for

下面是某些更多的实现细节. 我们决不要明确地形成 P_i 而仅仅做乘法

$$(\boldsymbol{I}-2\boldsymbol{u}_i\boldsymbol{u}_i^{\mathrm{T}})\boldsymbol{A}(i:m,i:n)=\boldsymbol{A}(i:m,i:n)-2\boldsymbol{u}_i(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{A}(i:m,i:n)),$$

这样做代价较小. 为存放 $\boldsymbol{P}_i$ 只需要 $\boldsymbol{u}_i$ 或 $\tilde{\boldsymbol{u}}_i$ 及 $\|\tilde{\boldsymbol{u}}_i\|$. 这些量可存放在 A 的 i 列中; 事实上它不需要改变!. 从而 QR 可以"覆盖"在 $\boldsymbol{A}$ 上, 其中 Q 以因子形式 $\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-1}$ 存放, 并且 $\boldsymbol{P}_i$ 被看作 $\tilde{\boldsymbol{u}}_i$ 存放在 $\boldsymbol{A}$ 的第 i 列对角线下面. (因为对角元被 $\boldsymbol{R}_{ii}$ 占用, 所以需要一个额外的长度为 n 的数组以存放 $\tilde{\boldsymbol{u}}_i$ 顶端的元素.)

回顾利用 $\boldsymbol{A}=\boldsymbol{QR}$ 求解最小二乘问题 $\min\|\boldsymbol{Ax}-\boldsymbol{b}\|_2$, 我们需要计算 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}$. 做法如下: $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}=\boldsymbol{P}_n\boldsymbol{P}_{n-1}\cdots\boldsymbol{P}_1\boldsymbol{b}$, 故只需要用 $\boldsymbol{P}_1,\boldsymbol{P}_2,\cdots,\boldsymbol{P}_n$ 继续不断地乘 $\boldsymbol{b}$:

for $i=1$ to n

$\quad \gamma=-2\cdot\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{b}(i:m)$

$\quad \boldsymbol{b}(i:m)=\boldsymbol{b}(i:m)+\gamma\boldsymbol{u}_i$

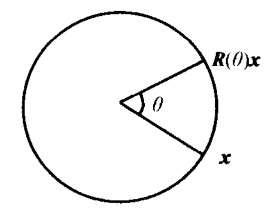
end for

代价是 n 个点积 $\gamma=-2\cdot\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{b}$ 和 n 个"saxpys" $\boldsymbol{b}+\gamma\boldsymbol{u}_i$. 这个方法计算 $\boldsymbol{A}=\boldsymbol{QR}$ 的代价是 $2n^2m-\frac{2}{3}n^3$, 给定 $\boldsymbol{QR}$ 后继的求解最小二乘问题的代价只是外加 $O(mn)$.

利用 $\boldsymbol{QR}$ 求解最小二乘问题的 LAPACK 程序是 sgels. 正如高斯消元法可使用矩阵-矩阵乘法和 3 级 BLAS(见 2.6 节)重组, 对 QR 分解同样可以这样做; 见问题 3.17. 在 Matlab 中, 若 $m\times n$ 阶矩阵 $\boldsymbol{A}$ 的行数大于列数, $\boldsymbol{b}$ 是 $m\times1$ 阶矩阵, 则用 $\boldsymbol{A}\backslash\boldsymbol{b}$ 解最小二乘问题. QR 分解本身通过 $[\boldsymbol{Q},\boldsymbol{R}]=\mathrm{qr}(\boldsymbol{A})$ 也是可以得到的.

3.4.2 吉文斯旋转

121 吉文斯旋转 $R(\theta)\equiv\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}$ 将任意向量 $\boldsymbol{x}\in\mathbb{R}^2$ 按逆时针方向旋转 θ:



也需要定义在坐标 i 和 j 中的吉文斯旋转 θ:

$$
\boldsymbol{R}(i,j,\theta) \equiv \begin{array}{c} \\ \\ \\ i \\ \\ j \\ \\ \\ \\ \end{array} \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & \ddots & & & & & & \\ & & & \cos\theta & & -\sin\theta & & & \\ & & & & \ddots & & & & \\ & & & \sin\theta & & \cos\theta & & & \\ & & & & & & \ddots & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix}
$$

(列标 i 在 $\cos\theta$ 所在列上方，j 在 $-\sin\theta$ 所在列上方)

给定 x，i 和 j，可选取 $\cos\theta$ 和 $\sin\theta$ 使得 x_j 为零，从而

$$
\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} \sqrt{x_i^2 + x_j^2} \\ 0 \end{bmatrix}
$$

或 $\cos\theta = \dfrac{x_i}{\sqrt{x_i^2 + x_j^2}}$和 $\sin\theta = \dfrac{-x_j}{\sqrt{x_i^2 + x_j^2}}$.

利用吉文斯旋转的 QR 算法是和利用豪斯霍尔德反射类似的，但是第 i 列化零时，一次把一个元素化零(譬如从底部到顶部).

例 3.6 说明利用吉文斯旋转计算 5×4 阶矩阵 QR 分解中的两个中间步. 为了从

$$
\begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & x & x \end{bmatrix} \text{前进到} \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & o & x \\ o & o & o & x \end{bmatrix}
$$

我们做乘法

$$
\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & c & -s \\ & & & s & c \end{bmatrix} \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & x & x \end{bmatrix} = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & o & x \end{bmatrix}
$$

以及

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & c' & -s' & \\ & & s' & c' & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & o & x \end{bmatrix} \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & o & x \\ o & o & o & x \end{bmatrix}. \qquad \diamond$$

利用吉文斯旋转作 QR 分解的代价是利用豪斯霍尔德反射的两倍. 后面其他的应用将需要吉文斯旋转.

下面是某些实现细节. 正如我们利用豪斯霍尔德反射时用 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 覆盖 $\boldsymbol{A}$ 一样, 对吉文斯旋转可以同样处理. 利用某些诀窍, 存放描述变换元素为零的信息. 因为吉文斯旋转只化一个元素为零, 所以必须存放有关在那里旋转的信息. 此事做法如下: 设 $s=\sin\theta$ 和 $c=\cos\theta$. 若 $|s|<|c|$, 则存放 $s\cdot\mathrm{sign}(c)$, 否则存放 $\frac{\mathrm{sign}(s)}{c}$. 为从存放的值 (称它为 p) 重新获得 s 和 c, 做法如下: 若 $|p|<1$, 则 $s=p$ 和 $c=\sqrt{1-s^2}$; 否则 $c=\frac{1}{p}$ 和 $s=\sqrt{1-c^2}$. 我们不只是存放 s 和计算 $c=\sqrt{1-s^2}$, 是因为当 s 接近于 1 时, c 将被不准确地重构. 也注意到可以重新获得不是 s 和 c 就是 $-s$ 和 $-c$;这在实际中是令人满意的.

还有一种应用一系列吉文斯旋转的方法, 而且比上面描述的过程执行的浮点运算要少. 这些旋转称为*快速吉文斯旋转*(fast Givens rotation)[7,8,33]. 因为它们计算 QR 分解的效果仍然慢于豪斯霍尔德反射, 所以我们不更多地考虑它们.

3.4.3　正交矩阵的舍入误差分析

这个分析证明 QR 分解和将讨论的许多特征值算法和奇异值算法的向后稳定性.

引理 3.1　*设 P 是一个精确的豪斯霍尔德(或吉文斯)变换, $\widetilde{P}$ 是其浮点运算近似. 则*

$$\mathrm{fl}(\widetilde{\boldsymbol{P}}\boldsymbol{A})=\boldsymbol{P}(\boldsymbol{A}+\boldsymbol{E})\quad \|\boldsymbol{E}\|_2=O(\varepsilon)\cdot\|\boldsymbol{A}\|_2$$

以及

$$\mathrm{fl}(\boldsymbol{A}\widetilde{\boldsymbol{P}})=(\boldsymbol{A}+\boldsymbol{F})\boldsymbol{P}\quad \|\boldsymbol{F}\|_2=O(\varepsilon)\cdot\|\boldsymbol{A}\|_2.$$

证明的梗概　对计算和应用 $\widetilde{\boldsymbol{P}}$ 的公式应用通常的公式 $\mathrm{fl}(a\odot b)=(a\odot b)(1+\varepsilon)$. 见问题 3.16. □

[123]　简言之, 这个结论表示应用一次单个的正交阵是向后稳定的.

定理 3.5　*考虑对 $\boldsymbol{A}_0$ 应用一系列正交变换. 则计算的积是 $\boldsymbol{A}_0+\delta\boldsymbol{A}$ 的一个精确的正交变换, 其中 $\|\delta\boldsymbol{A}\|_2=O(\varepsilon)\|\boldsymbol{A}\|_2$. 换言之, 整个计算是向后稳定的:*

$$\mathrm{fl}(\widetilde{\boldsymbol{P}}_j\widetilde{\boldsymbol{P}}_{j-1}\cdots\widetilde{\boldsymbol{P}}_1\boldsymbol{A}_0\widetilde{\boldsymbol{Q}}_1\widetilde{\boldsymbol{Q}}_2\cdots\widetilde{\boldsymbol{Q}}_j)=\boldsymbol{P}_j\cdots\boldsymbol{P}_1(\boldsymbol{A}_0+\boldsymbol{E})\boldsymbol{Q}_1\cdots\boldsymbol{Q}_j$$

其中 $\|\boldsymbol{E}\|_2=j\cdot O(\varepsilon)\cdot\|\boldsymbol{A}\|_2$. 如引理 3.1 中一样, 这里 $\widetilde{\boldsymbol{P}}_i$ 和 $\widetilde{\boldsymbol{Q}}_i$ 是浮点正交阵而 $\boldsymbol{P}_i$ 和 $\boldsymbol{Q}_i$ 是精确的正交阵.

证明 设$\overline{P}_j \equiv P_j \cdots P_1, \overline{Q}_j \equiv Q_1 \cdots Q_j$. 我们想要证明$A_j \equiv \mathrm{fl}(\widetilde{P}_j A_{j-1} \widetilde{Q}_j) = \overline{P}_j (A+E_j) \overline{Q}_j$，对某个$\|E_j\|_2 = jO(\varepsilon)\|A\|_2$成立. 我们递归地使用引理3.1. 对$j=0$结论显然成立，现假定结论对$j-1$成立. 然后计算

$$\begin{aligned} B &= \mathrm{fl}(\widetilde{P}_j A_{j-1}) \\ &= P_j(A_{j-1}+E') \quad \text{由引理 3.1} \\ &= P_j(\overline{P}_{j-1}(A+E_{j-1})\overline{Q}_{j-1}+E') \quad \text{由归纳假设} \\ &= \overline{P}_j(A+E_{j-1}+\overline{P}_{j-1}^{\mathrm{T}} E' \overline{Q}_{j-1}^{\mathrm{T}})\overline{Q}_{j-1} \\ &\equiv \overline{P}_j(A+E'')\overline{Q}_{j-1}, \end{aligned}$$

其中

$$\begin{aligned} \|E''\|_2 &= \|E_{j-1}+\overline{P}_{j-1}^{\mathrm{T}} E' \overline{Q}_{j-1}^{\mathrm{T}}\|_2 \leqslant \|E_{j-1}\|_2 + \|\overline{P}_{j-1}^{\mathrm{T}} E' \overline{Q}_{j-1}^{\mathrm{T}}\|_2 \\ &= \|E_{j-1}\|_2 + \|E'\|_2 = jO(\varepsilon)\|A\|_2 \end{aligned}$$

因为$\|E_{j-1}\|_2 = (j-1)O(\varepsilon)\|A\|_2$和$\|E'\|_2 = O(\varepsilon)\|A\|_2$. 用$\widetilde{Q}_i$右乘以同样方法处理. □

3.4.4 为什么用正交矩阵

如果在定理3.5中用一系列非正交(nonorthogonal)矩阵代替正交矩阵相乘，让我们考察误差将如何增长. 设X是精确的非正交变换，而$\widetilde{X}$是其浮点近似. 则通常的矩阵乘法的浮点误差分析告诉我们

$$\mathrm{fl}(\widetilde{X}A) = XA + E = X(A + X^{-1}E) \equiv X(A+F),$$

其中$\|E\|_2 \leqslant O(\varepsilon)\|X\|_2 \cdot \|A\|_2$，故$\|F\|_2 \leqslant \|X^{-1}\|_2 \cdot \|E\|_2 \leqslant O(\varepsilon) \cdot \kappa_2(X) \cdot \|A\|_2$.

因而误差$\|E\|_2$被条件数$\kappa_2(X) \geqslant 1$放大. 在一个较大的乘积$\widetilde{X}_k \cdots \widetilde{X}_1 A \widetilde{Y}_1 \cdots \widetilde{Y}_k$中误差将被放大$\prod_i \kappa_2(X_i) \cdot \kappa_2(Y_i)$倍. 这个因子达到极小当且仅当所有的$X_i$和$Y_i$是正交的（或标量乘正交阵），此时因子为1. 124

3.5 秩亏最小二乘问题

迄今为止我们假定当极小化$\|Ax-b\|_2$时A为满秩的. 当A秩亏或“接近”于秩亏时会发生什么情况呢? 此类问题产生于许多实际情况，例如从噪声数据中提取信号，某些积分方程的解，数字图像恢复，计算逆拉普拉斯变换等等[141,142]. 这些问题是极为病态的，为使它们良态，需要对它们的解强加额外的条件. 通过对解强加额外的条件使一个病态问题变为良态称为正则化(regularization)，且在数值分析的其他领域中当产生病态问题时也是这样做的.

例如，下面的命题指出，若A确切地是秩亏时，则最小二乘解不是恰好唯一的.

命题3.1 设A是$m \times n$阶矩阵，$m \geqslant n$，$\mathrm{rank}\, A = r < n$. 则存在一组$n-r$维的向量$x$使$\|Ax-b\|_2$达到极小.

证明 设$Az=0$. 则当x使$\|Ax-b\|_2$达到极小时，$x+z$也使$\|Ax-b\|_2$达到极小. □

由于A元素中的舍入和计算期间的舍入，最常见的情况是A将有一个或更多个非常小的计算的奇异值，而不是一些精确地为零的奇异值. 下列命题指出此时唯一解很可能是非常大而且毫无疑问地对右端b中的误差十分敏感(也可见定理3.4).

命题3.2　设$\sigma_{\min}=\sigma_{\min}(A)$是$A$的最小奇异值. 假设$\sigma_{\min}>0$. 则

1. 若x极小化$\|Ax-b\|_2$，则$\|x\|_2\geqslant|u_n^{\mathrm{T}}b|/\sigma_{\min}$，其中$u_n$是$A=U\Sigma V^{\mathrm{T}}$中$U$的最后一列.

2. 把b改变为$b+\delta b$可把x改变为$x+\delta x$，其中$\|\delta x\|_2$像$\|\delta b\|_2/\sigma_{\min}$一样大.

换言之，若A是几乎秩亏的($\sigma_{\min}$是小的)，则解x是病态的且可能非常大.

证明　第1部分，$x=A^{+}b=V\Sigma^{-1}U^{\mathrm{T}}b$，故$\|x\|_2=\|\Sigma^{-1}U^{\mathrm{T}}b\|_2\geqslant|(\Sigma^{-1}U^{\mathrm{T}}b)_n|=|u_n^{\mathrm{T}}b|/\sigma_{\min}$. 第2部分，选择$\delta b$平行于$u_n$. □

我们通过指出如何正则化一个精确地秩亏最小二乘问题来开始正则化讨论：假如A是$m\times n$阶矩阵，其秩$r<n$. 在$(n-r)$维解空间中，我们寻找唯一的最小范数解. 这个解的特征由下列命题来刻画.

125

命题3.3　当A是精确的奇异阵时，使$\|Ax-b\|_2$极小化的x可以刻画如下. 设$A=U\Sigma V^{\mathrm{T}}$有秩$r<n$，记A的SVD为

$$A=[U_1,U_2]\begin{bmatrix}\Sigma_1 & 0\\ 0 & 0\end{bmatrix}[V_1,V_2]^{\mathrm{T}}=U_1\Sigma_1V_1^{\mathrm{T}},\tag{3.1}$$

其中Σ_1是$r\times r$阶非奇异矩阵，U_1和V_1有r列. 设$\sigma=\sigma_{\min}(\Sigma_1)$是$A$的最小非零奇异值. 则

1. 所有的解x可写成$x=V_1\Sigma_1^{-1}U_1^{\mathrm{T}}b+V_2z$，$z$是一个任意的向量.

2. 当$z=0$时解x精确地有极小范数$\|x\|_2$，此时$x=V_1\Sigma_1^{-1}U_1^{\mathrm{T}}b$且$\|x\|_2\leqslant\|b\|_2/\sigma$.

3. 把b改变为$b+\delta b$，极小范数解x至多可能改变$\|\delta b\|_2/\sigma$.

换言之，唯一的极小范数解x的范数和条件数与A的最小非零奇异值有关.

证明　选择$\tilde{U}$使$[U,\tilde{U}]=[U_1,U_2,\tilde{U}]$是一个$m\times m$阶正交阵. 则

$$\begin{aligned}\|Ax-b\|_2^2&=\|[U,\tilde{U}]^{\mathrm{T}}(Ax-b)\|_2^2\\&=\left\|\begin{bmatrix}U_1^{\mathrm{T}}\\ U_2^{\mathrm{T}}\\ \tilde{U}^{\mathrm{T}}\end{bmatrix}(U_1\Sigma_1V_1^{\mathrm{T}}x-b)\right\|_2^2=\left\|\begin{bmatrix}\Sigma_1V_1^{\mathrm{T}}x-U_1^{\mathrm{T}}b\\ U_2^{\mathrm{T}}b\\ \tilde{U}^{\mathrm{T}}b\end{bmatrix}\right\|_2^2\\&=\|\Sigma_1V_1^{\mathrm{T}}x-U_1^{\mathrm{T}}b\|_2^2+\|U_2^{\mathrm{T}}b\|_2^2+\|\tilde{U}^{\mathrm{T}}b\|_2^2.\end{aligned}$$

(1)当$\Sigma_1V_1^{\mathrm{T}}x=U_1^{\mathrm{T}}b$或$x=V_1\Sigma_1^{-1}U_1^{\mathrm{T}}b+V_2z$时$\|Ax-b\|_2$达到极小，因为对于一切$z$，$V_1^{\mathrm{T}}V_2z=0$.

(2)因为V_1和V_2的列相互正交，所以由毕达哥拉斯定理推出$\|x\|_2^2=$

$\|V_1\Sigma_1^{-1}U_1^{\mathrm{T}}b\|_2^2+\|V_2z\|_2^2$，此式当 $z=0$ 时达到极小.

(3)用 δb 改变 b，至多用 $\|V_1\Sigma_1^{-1}U_1^{\mathrm{T}}\delta b\|_2\leq\|\Sigma_1^{-1}\|_2\|\delta b\|_2=\|\delta b\|_2/\sigma$ 改变 x. □

命题3.3告诉我们当最小的非零(nonzero)奇异值不是太小时，极小范数解 x 是唯一的并且可能是良态的. 这是下节中讨论的实际算法的关键. 126

例3.7 假如我们正在做一种药物对血糖水平影响的医学研究. 从每个病人(从 i 到 m 编号)那里收集数据，记录其初始血糖水平($a_{i,1}$)，最后的血糖水平(b_i)，给药量($a_{i,2}$)和其他体格检查量，包括持续一周治疗的每天的体重(从 $a_{i,3}$ 到 $a_{i,9}$). 对每个病人全部有 $n<m$ 个测量的体格检查量. 我们的目的是给定 $a_{i,1}$ 到 $a_{i,n}$ 去预估 b_i，并且明确地表达这个问题为最小二乘问题 $\min_x\|Ax-b\|_2$. 计划利用 x 通过计算点积 $\sum_{k=1}^{n}a_{jk}x_k$ 去预估未来病人 j 的最后的血糖水平 b_j.

因为人们的体重一般每天不会有较大的变化，它很可能是矩阵 A 的第3列到第9列，这些包含体重的列是非常相似的. 出于对自变量的兴趣，假如第3列和第4列是完全相同的(可能是体重被舍入到最近的磅的情况)，这意味着矩阵 A 是秩亏的且 $x_0=[0,0,1,-1,0,\cdots,0]^{\mathrm{T}}$ 是 A 的正确的零向量. 因此若 x 是最小二乘问题 $\min_x\|Ax-b\|_2$ 的(极小范数)解，则 $x+\beta x_0$ 也是一个(非极小范数)解，对任意的标量 β，包括 $\beta=0$ 和 $\beta=10^6$. 存在喜欢一个 β 值超过另一个 β 值的理由吗? 10^6 显然不是一个好的值，因为未来的病人 j 在第1天到第2天之间增加了一磅，将在最后血糖水平的预测值 $\sum_{k=1}^{n}a_{jk}x_k$ 中有一磅乘以 10^6 的差别. 选择 $\beta=0$，对应于极小范数解 x 是更加有道理的. ◇

关于秩亏问题利用极小范数解的进一步的论证是[141,142].

当 A 是非奇异方阵时，$Ax=b$ 的唯一解当然是 $x=A^{-1}b$. 若 A 的行数大于列数并且可能是秩亏的，则唯一的极小范数最小二乘解可类似地写成 $b=A^{+}b$，其中 Moore-Penrose 广义逆 A^{+} 定义如下：

定义3.2 (可能秩亏矩阵 A 的 Moore-Penrose 广义逆 A^{+})

如(3.1)式中那样设 $A=U\Sigma V^{\mathrm{T}}=U_1\Sigma_1V_1^{\mathrm{T}}$. 则 $A^{+}\equiv V_1\Sigma_1^{-1}U_1^{\mathrm{T}}$. 这也可写成 $A^{+}=V^{\mathrm{T}}\Sigma^{+}U$，其中 $\Sigma^{+}=\begin{bmatrix}\Sigma_1 & 0\\ 0 & 0\end{bmatrix}^{+}=\begin{bmatrix}\Sigma_1^{-1} & 0\\ 0 & 0\end{bmatrix}$.

所以最小二乘问题的解总是 $x=A^{+}b$，并且当 A 秩亏时 x 有极小范数.

3.5.1 用SVD解秩亏最小二乘问题

127

尽管有舍入误差，我们的目标还是计算极小范数解 x. 在上节中看到极小范数解是唯一的并且有一个与最小非零奇异值有关的条件数. 所以计算极小范数解需要知道最小非零奇异值，因此也需要知道 A 的秩. 主要的困难是矩阵的秩作为一个矩阵的函数不连续地改变.

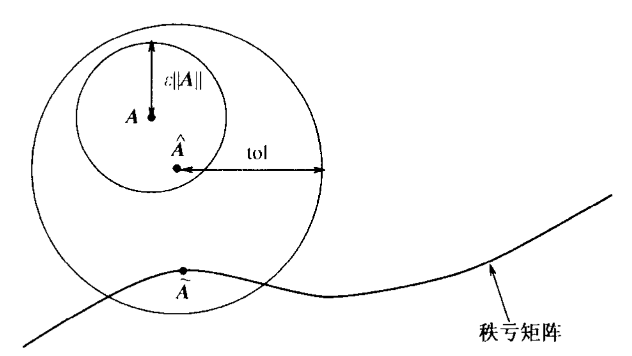
例如，2×2 阵 $A=\mathrm{diag}(1,0)$ 确实是奇异的，且它的最小非零奇异值是 $\sigma=1$. 如

命题3.3所述，$\min_x \|Ax-b\|_2$ 对 $b=[1,1]^T$ 的极小范数最小二乘解是 $x=[1,0]^T$，条件数 $1/\sigma=1$. 但当我们作一个任意微小的扰动得到 $\hat{A}=\text{diag}(1,\varepsilon)$，则 σ 下降到 ε 且 $x=[1,1/\varepsilon]^T$ 像它的条件数 $1/\varepsilon$ 一样变得非常大. 一般地，舍入将产生这样量级为 $O(\varepsilon)\|A\|_2$ 的微小扰动. 正如刚才所见，这个扰动能把条件数从 $1/\sigma$ 增加到 $1/\varepsilon$.

我们将如下处理这个算法的不连续性. 一般说来每个计算的奇异值 $\hat{\sigma}_i$ 满足 $|\hat{\sigma}_i-\sigma_i|\leqslant O(\varepsilon)\|A\|_2$. 这是向后稳定性的一个结果：计算的SVD应该是稍为不同的矩阵：$\hat{A}=\hat{U}\hat{\Sigma}\hat{V}^T=A+\delta A$ 的精确的SVD，$\|\delta A\|=O(\varepsilon)\cdot\|A\|$.（这在第5章中将详尽地讨论.）这意味着任意的 $\hat{\sigma}_i\leqslant O(\varepsilon)\|A\|_2$ 可视为零，因为舍入误差使它与0不能区别. 在上面 2×2 阶矩阵的例中，这意味着在求解最小二乘问题之前应该把 $\hat{A}$ 中的 ε 置为0. 这将把最小非零奇异值从 ε 提高到1，并且相应地把条件数从 $1/\varepsilon$ 下降到 $1/\sigma=1$.

更一般地，设tol是一个用户提供的数据 A 中的不确定性的度量. 舍入隐含着 $\text{tol}\geqslant\varepsilon\cdot\|A\|$，但是它可能较大，取决于 A 中数据的来源. 若 $\tilde{\sigma}_i>\text{tol}$，置 $\tilde{\sigma}_i=\hat{\sigma}_i$，否则置 $\tilde{\sigma}_i=0$. 设 $\tilde{\Sigma}=\text{diag}(\tilde{\sigma}_i)$. 我们称 $\tilde{U}\tilde{\Sigma}\tilde{V}^T$ 为 A 的截断SVD，因为已经把比tol小的奇异值置为0. 现在利用截断SVD代替原来的SVD求解最小二乘问题. 这是正确的，因为 $\|\hat{U}\tilde{\Sigma}\hat{V}^T-\hat{U}\hat{\Sigma}\hat{V}^T\|_2=\|\hat{U}(\tilde{\Sigma}-\hat{\Sigma})\hat{V}^T\|_2<\text{tol}$，即通过把每个 $\hat{\sigma}_i$ 改变为 $\tilde{\sigma}_i$ 引起 A 中的改变小于数据中用户固有的不确定性. 利用 $\tilde{\Sigma}$ 代替 $\hat{\Sigma}$ 的原因是在 $\hat{\Sigma}$ 的距离tol之内的所有矩阵中，$\tilde{\Sigma}$ 使最小非零奇异值 σ 达到最大. 换言之，它使极小范数最小二乘解 x 的范数及其条件数同时达到极小. 下图说明输入矩阵 A，$\hat{A}=\hat{U}\hat{\Sigma}\hat{V}^T$ 和 $\tilde{A}=\hat{U}\tilde{\Sigma}\hat{V}^T$ 之间的几何关系，这里我们把每个矩阵看作欧几里得空间 $\mathbb{R}^{m\cdot n}$ 中的一个点. 在此空间中，秩亏矩阵构成一个下面指出的曲面.

128



例3.8　对 20×10 阶秩亏矩阵 A_1（秩 $r_1=5$）和 A_2（秩 $r_2=7$）说明上面的过程. 记 A_1 或 A_2 的SVD为 $A_i=U_i\Sigma_iV_i^T$，其中 U_i，Σ_i 和 V_i 的公共的维数为 A_i 的秩 r_i. 这与命题3.3中的记号是一致的. A_i 的 r_i 个非零奇异值（Σ_i 的奇异值）在图3-4（对 A_1）和图3-5（对 A_2）中表为×. 注意在图3-4中的 A_1 有5个大的非零奇异值（全部值都稍稍超过1，故总的看来被标示在上部，在图的右边上），而图3-5中 A_2 的7个非零奇异值变化范围降到 $1.2\cdot10^{-9}\approx\text{tol}$.

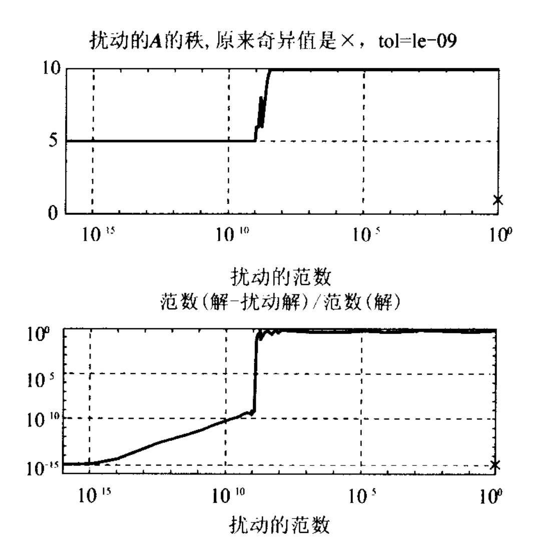


图 3-4 利用 $\mathrm{tol}=10^{-9}$、$\min_{y_1}\|(\boldsymbol{A}_1+\delta\boldsymbol{A})y_1-\boldsymbol{b}_1\|_2$ 的截断最小二乘解的曲线图，$\boldsymbol{A}_1$ 的奇异值用 × 表示. 范数 $\|\delta\boldsymbol{A}\|_2$ 是水平轴. 上面的曲线图绘出 $\boldsymbol{A}_1+\delta\boldsymbol{A}$ 的秩，即超过 tol 的奇异值数目. 下面的曲线图绘出 $\|\boldsymbol{y}_1-\boldsymbol{x}_1\|_2/\|\boldsymbol{x}_1\|_2$，其中 $\boldsymbol{x}_1$ 是关于 $\delta\boldsymbol{A}=0$ 的解

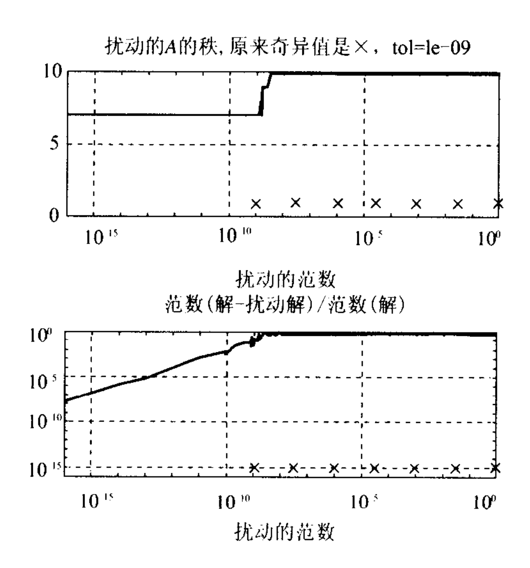


图 3-5 利用 $\mathrm{tol}=10^{-9}$, $\min_{y_2}\|(\boldsymbol{A}_2+\delta\boldsymbol{A})\boldsymbol{y}_2-\boldsymbol{b}_2\|_2$ 的截断最小二乘解的曲线图. $\boldsymbol{A}_2$ 的奇异值用 × 表示. 范数 $\|\delta\boldsymbol{A}\|_2$ 是水平轴. 上面的曲线图绘出 $\boldsymbol{A}_2+\delta\boldsymbol{A}$ 的秩，即超过 tol 的奇异值数目. 下面的曲线图给出 $\|\boldsymbol{y}_2-\boldsymbol{x}_2\|_2/\|\boldsymbol{x}\|_2$，其中 $\boldsymbol{x}_2$ 是关于 $\delta\boldsymbol{A}=0$ 的解

然后选择一个 r_i 维向量 $\boldsymbol{x}'_i$ 并设 $\boldsymbol{x}_i=\boldsymbol{V}_i\boldsymbol{x}'_i$ 和 $\boldsymbol{b}_i=\boldsymbol{A}_i\boldsymbol{x}_i=\boldsymbol{U}_i\sum_i\boldsymbol{x}_i'$，故 $\boldsymbol{x}_i$ 正好是使 $\|\boldsymbol{A}_i\boldsymbol{x}_i-\boldsymbol{b}_i\|_2$ 达到极小的极小范数解. 接着考虑一系列扰动问题 $\boldsymbol{A}_i+\delta\boldsymbol{A}$，其中扰动 $\delta\boldsymbol{A}$ 被随机地选择具有一个范数的变动范围，并利用截断最小二乘过程以 tol $=10^{-9}$ 解最小二乘问题 $\|(\boldsymbol{A}_i+\delta\boldsymbol{A})\boldsymbol{y}_i-\boldsymbol{b}_i\|_2$. 图3-4 和3-5 中的曲线标示 $\boldsymbol{A}_i+\delta\boldsymbol{A}$ 的计算秩（计算的奇异值数目超过 tol $=10^{-9}$）对 $\|\delta\boldsymbol{A}\|_2$（在上面的图中）和对误差 $\|\boldsymbol{y}_i-\boldsymbol{x}_i\|_2/\|\boldsymbol{x}_i\|_2$ 的关系曲线（在下面的图中）. 产生这些图的 Matlab 代码在 HOMEPAGE/Matlab/RankDeficient. m 中.

图3-4 是最简单的情况，故首先考虑它. $\boldsymbol{A}_1+\delta\boldsymbol{A}$ 有 5 个接近于 1 或稍超过 1 的奇异值，并且其他 5 个等于或小于 $\|\delta\boldsymbol{A}\|_2$. 对 $\|\delta\boldsymbol{A}\|_2<\text{tol}$，$\boldsymbol{A}_1+\delta\boldsymbol{A}$ 的计算秩保持与 $\boldsymbol{A}_1$ 的秩一样，即等于5. 误差也从接近机器精度（$\approx10^{-16}$）缓慢地增加到 10^{-10}，接近 $\|\delta\boldsymbol{A}\|_2=\text{tol}$，然后对大的 $\|\delta\boldsymbol{A}\|_2$，秩和误差同时分别地跳跃到 10 和 1. 这与命题 3.3 中的分析是一致的，这说明条件数是最小非零奇异值的倒数，即最小奇异值超过 tol. 对 $\|\delta\boldsymbol{A}\|_2<\text{tol}$，这个最小的非零奇异值接近 1 或稍超过 1. 所以命题 3.3 预测 $\|\delta\boldsymbol{A}\|_2/O(1)=\|\delta\boldsymbol{A}\|_2$ 的一个误差. 这个良态情况由图 3-4 的下面的曲线图中 $\|\delta\boldsymbol{A}\|_2=\text{tol}$ 的左边给出的小误差所证实. 另一方面，当 $\|\delta\boldsymbol{A}\|_2>\text{tol}$ 时，最小的非零奇异值是 $O(\|\delta\boldsymbol{A}\|_2)$，这是十分小的，正如图 3-4 下面的曲线图中 $\|\delta\boldsymbol{A}\|_2=\text{tol}$ 的右边所示，使误差跳跃到 $\|\delta\boldsymbol{A}\|_2/O(\|\delta\boldsymbol{A}\|_2)=O(1)$. 129

在图 3-5 中，$\boldsymbol{A}_2$ 的非零奇异值也用 × 表示. 最小的一个是 $1.2\cdot10^{-9}$，恰好大于 tol. 故当 $\|\delta\boldsymbol{A}\|_2<\text{tol}$ 时预测的误差是 $\|\delta\boldsymbol{A}\|_2/10^{-9}$，当 $\|\delta\boldsymbol{A}\|_2=\text{tol}$ 时，它增长到 $O(1)$. 这由图 3-5 中下面的曲线图所证实. ◇

3.5.2 用选主元的 QR 分解解秩亏最小二乘问题

一个代价较低的但有时准确度低于 SVD 的可供选择的办法，是选主元的 QR 分解（QR with pivoting）. 在精确的算术运算中，若 $\boldsymbol{A}$ 的秩 $r<n$ 且它的前 r 列是无关的，则它的 QR 分解的外表特征是

$$\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{R}=\boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}_{11} & \boldsymbol{R}_{12}\\ 0 & 0\\ 0 & 0\end{bmatrix}_r,$$

其中 $\boldsymbol{R}_{11}$ 是 $r\times r$ 阶非奇异矩阵而 $\boldsymbol{R}_{12}$ 是 $r\times(n-r)$ 阶矩阵. 由于舍入我们可能希望计算

$$\boldsymbol{R}=\begin{bmatrix}\boldsymbol{R}_{11} & \boldsymbol{R}_{12}\\ 0 & R_{22}\\ 0 & 0\end{bmatrix}$$

其中 $\|\boldsymbol{R}_{22}\|_2$ 非常小，靠近 $\varepsilon\|\boldsymbol{A}\|_2$ 阶. 此时正好能够置 $\boldsymbol{R}_{22}=0$ 并极小化 $\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2$ 如下：设 $[\boldsymbol{Q},\tilde{\boldsymbol{Q}}]$ 是方的正交阵使得

$$\|Ax-b\|_2^2=\left\|\begin{bmatrix}Q^{\mathrm T}\\ \widetilde{Q}^{\mathrm T}\end{bmatrix}(Ax-b)\right\|_2^2=\left\|\begin{bmatrix}Rx-Q^{\mathrm T}b\\ -\widetilde{Q}^{\mathrm T}b\end{bmatrix}\right\|_2^2$$

$$=\|Rx-Q^{\mathrm T}b\|_2^2+\|\widetilde{Q}^{\mathrm T}b\|_2^2.$$

记 $Q=[Q_1,Q_2]$ 和 $x=\begin{bmatrix}x_1\\ x_2\end{bmatrix}$ 与 $R=\begin{bmatrix}R_{11} & R_{12}\\ 0 & 0\end{bmatrix}$ 保持一致，通过选择 $x=\begin{bmatrix}R_{11}^{-1}(Q_1^{\mathrm T}b-R_{12}x_2)\\ x_2\end{bmatrix}$，对任意的 x_2，使得

$$\|Ax-b\|_2^2=\|R_{11}x_1+R_{12}x_2-Q_1^{\mathrm T}b\|_2^2+\|Q_2^{\mathrm T}b\|_2^2+\|\widetilde{Q}^{\mathrm T}b\|_2^2$$

达到极小. 注意选择 $x_2=0$ 不一定极小化 $\|x\|_2$，但是特别当 R_{11} 良态且 $R_{11}^{-1}R_{12}$ 是小的时候，这是一个适当的选择.

遗憾的是，这个方法不是可靠的，因为即使没有 R_{22} 是小的，R 也可能是几乎秩亏的. 例如，$n\times n$ 阶上双对角阵

$$A=\begin{bmatrix}\frac{1}{2} & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1\\ & & & \frac{1}{2}\end{bmatrix}$$

130 ~ 131

有 $\sigma_{\min}(A)\approx 2^{-n}$，但 $A=Q\cdot R$，其中 $Q=I,R=A$，且没有 R_{22} 是小的.

要处理这个辨认秩亏的失败，我们可做*列选主元*(column pivoting)的 QR 分解. 这意味着分解 $AP=QR$,P 是一个置换阵. 这个想法是在第 i 步(列的数目，范围从 1 到 n)从 A 的未完成的部分(第 i 列到第 n 列和第 i 行到第 m 行)选择最大范数的列并把它同第 i 列交换. 然后进行计算通常的豪斯霍尔德变换使第 i 列的第 $i+1$ 个到第 m 个元素化零. 这种选主元策略试图尽可能保持 R_{11} 良态和使 R_{22} 尽可能小.

例 3.9 若对上例(0.5 在对角线上,1 在上对角线上)用列选主元计算 QR 分解，对 $n=11$，得到 $R_{11,11}=4.23\cdot 10^{-4}$，这是对 $\sigma_{\min}(A)=3.66\cdot 10^{-4}$ 的一个适当的近似. 注意 $R_{nn}\geq\sigma_{\min}(A)$,因为 $\sigma_{\min}(A)$ 是能够降秩的最小扰动的范数，于是置 R_{nn} 为 0 来降秩. ◇

人们只能指出 $\dfrac{R_{nn}}{\sigma_{\min}(A)}\leq 2^n$，但通常 R_{nn} 是 $\sigma_{\min}(A)$ 的一个适当的近似. 然而，最坏的情况是像 GEPP 中最坏的情况主元增长那样坏.

比列选主元 QR 分解更精致的选主元格式，称为*秩展现*(rank-revealing)QR 算法，曾经是一个最新的研究题目. 检测秩的秩展现 QR 算法更为可靠，并且有时也快于已

经作了讨论的列选主元 QR 算法. [28,30,48,50,109,126,128,150,196,236]. 在下一节中进一步讨论它们.

列选主元的 QR 分解可以利用 LAPACK 中的子程序 sgeqpf. LAPACK 也有几个类似的分解可以利用: RQ(sgerqf), LQ(sgelqf)和 QL(sgeqlf). 未来的 LAPACK 版本将包含 QR 的改进形式.

3.6 最小二乘问题解法的性能比较

求解稠密最小二乘问题最快的算法是什么? 如3.2 节中讨论的那样, 解正规方程是最快的, 接着是 QR 和 SVD. 若 $\boldsymbol{A}$ 是十分良态的, 则正规方程几乎像其他方法一样精确, 因此即使正规方程不是数值稳定的, 同样可使用它们. 当 $\boldsymbol{A}$ 不是良态的但决不秩亏时, 我们应该使用 QR 算法.

因为秩亏最小二乘问题快速算法的设计属于当前的研究领域, 所以推荐使用一个单独的算法是困难的. 我们概述一个最新的研究[206], 比较几种算法的性能, 把 132 它们与非秩亏情况最快的稳定算法作比较: 利用3.4.1 节中描述的豪斯霍尔德变换执行的不选主元的 QR 算法, 具有问题 3.17 中描述的存储器分层优化. 这些比较在 IBM RS6000/590 上以双精度算术运算作出. 在比较中包括的是 3.5.2 节中提及的秩展现 QR 算法以及 SVD 的各种执行过程(见 5.4 节). 测试了各种大小阶数和有各种奇异值分布的矩阵. 我们介绍两种奇异值分布的结果:

类型 1: 随机矩阵, 其中每个元素是从 -1 到 1 均匀分布的.

类型2: 奇异值从 1 到 ε 几何分布的矩阵(换言之, 第 i 个奇异值是 γ^i, 其中 γ 使得 $\gamma^n=\varepsilon$).

类型 1 矩阵一般是良态的, 而类型 2 矩阵是秩亏的. 我们测试小的方阵($n=m=20$)和大的方阵($m=n=1600$). 测试方阵是因为在 $m\times n$ 阶矩阵 $\boldsymbol{A}$ 中当 m 充分大于 n 时, 做一次 QR 分解作为一个"预处理步"代价较低, 然后对 $\boldsymbol{R}$ 执行秩展现 QR 或 SVD(这是在 LAPACK 中做的). 若 $m\gg n$, 则初始的 QR 分解的代价超过对 $n\times n$ 阶矩阵 $\boldsymbol{R}$ 后继运算的代价, 而所有的算法代价几乎相同.

秩展现 QR 算法最快的形式是[30,196]中的那个形式. 对类型 1 矩阵, 这个算法变化范围从慢于不选主元的 QR 算法 3.2 倍(对 $n=m=20$)到仅仅 1.1 倍(对 $n=m=1600$). 对类型 2 矩阵, 它的变化范围从慢 2.3 倍(对 $n=m=20$)到慢 1.2 倍(对 $n=m=1600$). 与之对照, 现今的 LAPACK 算法 dgeqpf 对两类矩阵慢 2 倍到2.5 倍.

SVD 算法最快的形式是[58]中的那个形式, 虽然对 $n=m=1600$ 基于分而治之的一种算法(见 5.3.3 节)几乎同样地快. (基于分而治之的一种算法还使用较小的存储器.)对类型 1 矩阵 SVD 算法从慢 7.8 倍(对 $n=m=20$)到慢 3.3 倍(对 $n=m=1600$). 对类型 2 矩阵, SVD 算法从慢 3.5 倍(对 $n=m=20$)到慢 3.0 倍(对 $n=m=1600$). 与之相比, 现行的 LAPACK 算法 dgelss 变化范围从慢 4 倍(对类型 2 矩阵, $n=m=$

20)到慢97倍(对类型1矩阵，$n=m=1600$). 这个巨大的减速明显地是由于存储器分层结构影响.

于是，我们看到在解秩亏最小二乘问题中可靠性和速度之间存在一个交替换位，不选主元的QR算法最快但最不可靠，SVD最慢但最可靠，而秩展现QR算法居中. 若$m \gg n$，则所有算法的代价几乎相同. 算法的选择依赖于速度和可靠性对用户而言的相对重要性.

未来的LAPACK版本对最小二乘问题将同时包含秩展现QR和SVD算法两个改进的形式.

133

3.7 第3章的参考书目和其他话题

关于最小二乘问题最好的最新的参考书是[33]，该书还讨论了本书讨论的基本问题的变形(例如约束的、加权的和更新的最小二乘问题)，正则化秩亏问题的不同的方法和稀疏最小二乘问题的软件. 也可见[121]的第5章和[168]. 最小二乘问题的扰动理论和误差界在[149]中作了详尽的讨论. 秩展现的QR分解在[28,30,48,50,126,150,196,206,236]中作了讨论. 特别地，这些论文考察秩确定中的代价和精度之间的交替换位，并且在[206]中，对秩亏最小二乘问题可利用的方法有一个综合性的性能比较.

3.8 第3章问题

问题3.1(容易) 通过证明r_{ji}的两个公式按精确的算术运算得到相同的结果，来证明算法3.1的两个变形CGS和MGS在数学上是等价的.

问题3.2(容易) 这个问题将说明在计算一个矩阵的QR分解的三个算法：豪斯霍尔德QR(算法3.2)、CGS(算法3.1)和MGS(算法3.1)中间数值稳定性的差别. 可从HOMEPAGE/Matlab/QRStability.m得到Matlab程序QRStability.m. 这个程序按用户指定的维数m和n以及条件数cnd生成随机矩阵，利用三个算法计算它们的QR分解，并度量结果的精度. 一个稳定的算法残差$\|\boldsymbol{A}-\boldsymbol{Q}\cdot\boldsymbol{R}\|/\|\boldsymbol{A}\|$应该在机器精度$\varepsilon$附近. Q的正交性$\|\boldsymbol{Q}^{\mathrm{T}}\cdot\boldsymbol{Q}-\boldsymbol{I}\|$也应该在$\varepsilon$附近. 对小的矩阵维数(例如m=6和n=4)，中等数量的随机矩阵(samples=20)，和变化范围从cnd=1直到cnd=10^{15}的条件数运行这个程序. 描述你所看到的情况. 哪种算法比其他的算法更稳定？看看你能描述多大的$\|\boldsymbol{Q}^{\mathrm{T}}\cdot\boldsymbol{Q}-\boldsymbol{I}\|$能作为选择算法，cnd和$\varepsilon$的函数.

问题3.3(中等；困难) 设$\boldsymbol{A}$是$m\times n$阶矩阵，$m\geq n$且满秩.

1.(中等) 证明$\begin{bmatrix}\boldsymbol{I} & \boldsymbol{A}\\ \boldsymbol{A}^{\mathrm{T}} & 0\end{bmatrix}\cdot\begin{bmatrix}r\\ x\end{bmatrix}=\begin{bmatrix}b\\ 0\end{bmatrix}$有一个解，其中$x$极小化$\|\boldsymbol{A}x-\boldsymbol{b}\|_2$. 用这个形式表示的一个原因是当我们希望一个更精确的解时，可以对这个线性方程组应

用迭代精化(见2.5节).

2.(中等)　用$\boldsymbol{A}$的奇异值来表示，系数矩阵的条件数是什么？提示：利用$\boldsymbol{A}$的SVD.

[134]

3.(中等)　按照一个2×2块矩阵，给出系数阵之逆阵的显式表达式. 提示：利用2×2块高斯消元法. 我们前面已经看到的(2,1)块元素在哪里？

4.(困难)　说明为改善x的精度如何利用$\boldsymbol{A}$的QR分解去实现迭代精化算法.

问题3.4(中等)　加权最小二乘(weighted least squares)：若$\boldsymbol{A}x-\boldsymbol{b}$的某些分量比另外的分量更重要，则可以用一个标量$d_i$对它们加权并用解加权最小二乘问题$\min\|\boldsymbol{D}(\boldsymbol{A}x-\boldsymbol{b})\|_2$代替，其中$\boldsymbol{D}$有对角元$d_i$. 更一般地，记得若$\boldsymbol{C}$是对称正定阵，则$\|x\|_C\equiv(x^{\mathrm{T}}\boldsymbol{C}x)^{1/2}$是一个范数，可以考虑极小化$\|\boldsymbol{A}x-\boldsymbol{b}\|_C$. 对这个问题导出正规方程以及对应于前面的问题的公式.

问题3.5(中等；Z. Bai)　设$\boldsymbol{A}\in\mathbb{R}^{n\times n}$是正定的. 若$\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}_2=0$，则称两个向量$\boldsymbol{u}_1$和$\boldsymbol{u}_2$是$\boldsymbol{A}$-正交的. 若$\boldsymbol{U}\in\mathbb{R}^{n\times r}$且$\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{U}=\boldsymbol{I}$，则$\boldsymbol{U}$的列称为是$\boldsymbol{A}$-标准正交的. 证明每个子空间有一个$\boldsymbol{A}$-标准正交基.

问题3.6(容易；Z. Bai)　设$\boldsymbol{A}$有形式

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{S}\end{bmatrix},$$

其中$\boldsymbol{R}$是$n\times n$上三角阵，$\boldsymbol{S}$是$m\times n$稠密阵. 描述用豪斯霍尔德变换把$\boldsymbol{A}$化为上三角形的一个算法. 你的算法应该不“填补”$\boldsymbol{R}$中的零，因而应该比算法3.2应用$\boldsymbol{A}$需要较少的运算.

问题3.7(中等；Z. Bai)　若$\boldsymbol{A}=\boldsymbol{R}+\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}$，其中$\boldsymbol{R}$是上三角阵，$\boldsymbol{u}$和$\boldsymbol{v}$是列向量，描述一个计算$\boldsymbol{A}$的QR分解的有效算法. 提示：利用吉文斯旋转，你的算法应该进行$O(n^2)$次运算. 与之对照，算法3.2进行了$O(n^3)$次运算.

问题3.8(中等；Z. Bai)　设$\boldsymbol{x}\in\mathbb{R}^n$并设$\boldsymbol{P}$是一个豪斯霍尔德阵使得$\boldsymbol{P}x=\pm\|x\|_2e_1$. 设$\boldsymbol{G}_{1,2},\cdots,\boldsymbol{G}_{n-1,n}$是吉文斯旋转且设$Q=G_{1,2}\cdots G_{n-1,n}$. 假如$Qx=\pm\|x\|_2e_1$. $\boldsymbol{P}$一定等于$\boldsymbol{Q}$吗？(你需要给出一个证明或反例)

问题3.9(容易；Z. Bai)　设$\boldsymbol{A}$是$m\times n$阶矩阵，其SVD为$\boldsymbol{A}=\boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}}$. 依据$\boldsymbol{U}$、$\Sigma$和$\boldsymbol{V}$计算下列矩阵的SVD：

[135]

1. $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}$；　2. $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}$；　3. $\boldsymbol{A}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}$；　4. $\boldsymbol{A}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}$.

问题3.10(中等；R. Schreiber)　正如定理3.3的第9部分中所定义的，设$\boldsymbol{A}_k$是$\boldsymbol{A}$的最佳秩k近似. 设σ_i是$\boldsymbol{A}$的第i个奇异值. 证明若$\sigma_k>\sigma_{k+1}$则$\boldsymbol{A}_k$是唯一的.

问题3.11(容易；Z. Bai)　设$\boldsymbol{A}$是$m\times n$阶矩阵. 证明取遍所有$n\times m$阶矩阵$\boldsymbol{X}$使$\|\boldsymbol{A}\boldsymbol{X}-\boldsymbol{I}\|_F$达到极小的是$\boldsymbol{X}=\boldsymbol{A}^+$(Moore-Penrose广义逆).

问题3.12(中等；Z. Bai)　设$\boldsymbol{A}$、$\boldsymbol{B}$和$\boldsymbol{C}$是具有适当阶数的矩阵，使乘积$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{B}^{\mathrm{T}}$有意义. 设$\chi$是使$\|\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}-\boldsymbol{C}\|_F$达到极小的矩阵$\boldsymbol{X}$的集合，并设$\boldsymbol{X}_0$是$\chi$中使$\|\boldsymbol{X}\|_F$达到极小的唯一的矩阵. 证明$\boldsymbol{X}_0=\boldsymbol{A}^+\boldsymbol{C}\boldsymbol{B}^+$. 提示：利用$\boldsymbol{A}$和$\boldsymbol{B}$的SVD.

问题3.13(中等；Z. Bai) 证明A的Moore-Penrose广义逆满足下列恒等式:

$$AA^+A=A,$$
$$A^+AA^+=A^+,$$
$$A^+A=(A^+A)^{\mathrm{T}},$$
$$AA^+=(AA^+)^{\mathrm{T}}.$$

问题3.14(中等) 证明定理3.3的第4部分：设$H=\begin{bmatrix}0 & A^{\mathrm{T}}\\ A & 0\end{bmatrix}$，其中$A$是方阵而$A=U\Sigma V^{\mathrm{T}}$是它的SVD. 设$\Sigma=\mathrm{diag}(\hat{\sigma}_1,\cdots,\sigma_n)$，$U=[u_1,\cdots,u_n]$，$V=[v_1,\cdots,v_n]$. 证明$H$的对应于单位特征向量$\frac{1}{\sqrt{2}}\begin{bmatrix}v_i\\ \pm u_i\end{bmatrix}$的$2n$个特征值是$\pm\sigma_i$. 把上述结论推广至长方阵$A$的情况.

问题3.15(中等) 设A是$m\times n$阶满秩阵，$m<n$. 则$\min\|Ax-b\|_2$称为亚定最小二乘问题. 证明解是一个$n-m$维集合. 利用适当修改的正规方程、QR分解和SVD计算唯一的极小范数解.

问题3.16(中等) 证明引理3.1.

136

问题3.17(困难) 在2.6.3节中，曾指出在每步为了利用2级BLAS和3级BLAS运算的高速度来如何重组高斯消元法. 在本题中将指出如何应用一系列使用2级和3级BLAS的豪斯霍尔德变换.

1. 设$u_1,\cdots,u_b$是一系列维数为n的向量，其中$\|u_i\|_2=1$且u_i的前$i-1$个分量为0. 设$P=P_b\cdot P_{b-1}\cdots P_1$，其中$P_i=I-2u_iu_i^{\mathrm{T}}$是豪斯霍尔德变换. 证明存在一个$b\times b$阶下三角阵$T$使得$P=I-UTU^{\mathrm{T}}$，其中$U=[u_1,\cdots,u_b]$. 特别地，提供一个计算$T$的元素的算法. 这个恒等式表明可以用$U$，$T$和$U^{\mathrm{T}}$的三个矩阵乘法(加上计算$T$的代价)代替$P_1$到$P_b$的$b$个豪斯霍尔德变换的乘法.

2. 设House(x)是一个向量x的函数，它获得一个单位向量u使得$(I-2uu^{\mathrm{T}})x=\|x\|_2e_1$；我们指出如何实施3.4节中的House($x$). 于是计算$m\times n$阶矩阵$A$的QR分解可写成

```
for i = 1 : m
    u_i = House(A(i : m,i))
    P_i = I - 2u_i u_i^T
    A(i : m,i : n) = P_i A(i : m,i : n)
end for
```

指出如何依据2级BLAS以一种有效的方式(尤其是矩阵向量乘法和秩1更新)来执行这个程序. 浮点运算量是什么？(只要n和m中的高阶项就够了.)用上面同样的记号编写一个简短的程序就足够了. (虽然用Matlab试验而与Matlab本身的QR分解作比较是一个确保正确的好方法!)

3. 利用第1步的结果，指出如何依据3级BLAS来执行QR分解. 运算量是什么? 正如2.6节中加速高斯消元法一样，这个方法也用于加速QR分解. 它被用于LAPACK中程序 sgeqrf 中.

问题3.18(中等) 常常有兴趣考虑求解约束最小二乘问题(constrained least squares problem)，其中解 $\boldsymbol{x}$ 除了极小化 $\|\boldsymbol{Ax}-\boldsymbol{b}\|_2$ 外还必须满足一个线性或非线性约束. 下面我们考虑这样的一个问题. 假如希望选择 $\boldsymbol{x}$ 使 $\|\boldsymbol{Ax}-\boldsymbol{b}\|_2$ 达到极小且服从线性约束条件 $\boldsymbol{Cx}=\boldsymbol{d}$. 假定 $\boldsymbol{A}$ 是 $m\times n$ 阶矩阵，C 是 $p\times n$ 阶满秩阵. 还假定 $p\leqslant n$(故保证 $\boldsymbol{Cx}=\boldsymbol{d}$ 相容)且 $n\leqslant m+p$(故方程组不是亚定的). 证明在假设 $\begin{bmatrix}\boldsymbol{A}\\\boldsymbol{C}\end{bmatrix}$ 列满秩之下存在唯一解. 指出如何利用两个QR分解和某些矩阵乘法以及解某些三角形方程组去计算 $\boldsymbol{x}$. 提示：注意LAPACK程序 sgglse 及其在LAPACK手册中的描述[10](NETLIB/lapack/lug/lapack_lug.html).

137

问题3.19(困难；程序设计) (用Matlab或其他语言)编写一个程序如例3.3中描述的那样，利用最小二乘法更新大地测量数据库. 取一组"界标"的近似坐标 (x_i,y_i) 和一组新的角度值 θ_j 和距离值 L_{ij} 作为输入. 输出应该是每个界标的校正 $(\delta x_i,\delta y_i)$，校正的误差界以及老的和新的界标的图(三角剖分).

问题3.20(困难) 证明定理3.4.

问题3.21(中等) 利用3.5.1节的秩亏最小二乘方法，重做例3.1. 这个方法能改进高阶近似多项式的精度吗?

第 4 章　非对称特征值问题

4.1 概　　述

我们对单个非对称阵 $\boldsymbol{A}$ 的特征值问题讨论典范型(4.2 节)，扰动理论(4.3 节)和算法(4.4 节). 第 5 章专门讨论实对称矩阵 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$(和 SVD)的特殊情况 . 4.5 节讨论涉及多个矩阵的特征值问题的推广，包括由振动系统的分析、线性微分方程的解和计算几何引出的应用. 最后，4.6 节以列表形式总结所有的典范型、算法、代价、应用和可用软件.

可以粗略地把特征值问题的算法分成两类：*直接法*(direct method)和*迭代法*(iterative method). 本章只考虑直接法，它用来计算全部特征值和(自选的)特征向量. 直接法一般用于稠密阵，计算全部特征值和特征向量的代价为 $O(n^3)$ 次运算. 这个代价对实际的矩阵元素是相对不敏感的.

实际中使用的主要的直接法是*隐式位移的 QR 迭代*(QR iteration with implicit shift)(见 4.4.8 节). 有趣的是这个算法作了 30 多年可靠的贡献之后，近期发现了它的收敛性缺陷，并对算法作了分析和修补[25,65]. 即使当前的算法被认为是十分可靠的，但是也还没有整体收敛性证明. 因此设计一个数值稳定的、整体(且快速的)收敛的算法还是个有待解决的问题. [注意“直接”法还必须作迭代，因为求特征值数学上等价于求多项式的根，而后者不可能存在非迭代的方法. 如果经验表明，用一个固定的迭代次数收敛(几乎)决不失败，则我们称该方法是*直接的*.]

在第 7 章中讨论的*迭代法*通常应用于稀疏阵或者以最方便的运算去执行矩阵 - 向量乘法的矩阵. 迭代法一般只提供特征值和特征向量的一个子集的近似，并且只是为了得到几个足够精确的特征值而运行足够长时间，而不是大量的迭代次数. 它们的收敛性强烈地依赖于矩阵的元素.

139

4.2 典　范　型

定义 4.1　*多项式 $p(\lambda)=\det(\boldsymbol{A}-\lambda\boldsymbol{I})$ 称为 $\boldsymbol{A}$ 的特征多项式. $p(\lambda)=0$ 的根是 $\boldsymbol{A}$ 的特征值.*

因为特征多项式 $p(\lambda)$ 的次数等于 $\boldsymbol{A}$ 的维数 n，多项式有 n 个根，所以 $\boldsymbol{A}$ 有 n 个特征值.

定义4.2　满足$Ax=\lambda x$的非零向量x是特征值λ的一个(右)特征向量. 使得$y^*A=\lambda y^*$的非零向量y^*是左特征向量.（记得$y^*=(\bar{y})^T$是y的共轭转置.）

大多数算法将涉及到把矩阵A变换到较简单的形式或典范型(canonical)，根据这些形式容易计算A的特征值和特征向量. 这些变换称为相似变换(similarity transformation)(见下面). 两个最普通的典范型称为若尔当型和舒尔型(Schur form). 若尔当型理论上是有用的，但按数值稳定方式计算是非常困难的，这就是为什么我们的算法将针对计算舒尔型.

为明确若尔当型和舒尔型的目的，我们要问哪种矩阵具有其特征值容易计算的性质. 最容易的情况应当是对角阵其特征值就是它的对角元. 同样容易的应当是三角阵，其特征值也是它的对角元. 下面将看到一个若尔当型或舒尔型的矩阵是三角阵. 但是记得一个实矩阵可能有复特征值，因为它的特征多项式的根可能是实的或复的. 因为实三角阵只能有实的特征值，所以，不一定存在一个实的三角阵，它具有一般的实矩阵那样相同的特征值. 所以，必须或者使用复数或者寻找实矩阵的典范型之外的实三角矩阵. 结果变成考虑块三角阵(block triangular matrices)，即下列形式的矩阵

$$A=\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1b} \\ & A_{22} & \cdots & A_{2b} \\ & & \ddots & \vdots \\ & & & A_{bb} \end{bmatrix}, \tag{4.1}$$

140　就足够了，其中A_{ii}是方阵并且A_{ii}块下面所有的元素为零. 容易证明A的特征多项式$\det(A-\lambda I)$是A_{ii}的特征多项式之积$\prod_{i=1}^{b}\det(A_{ii}-\lambda I)$，所以$A$的特征值集$\lambda(A)$是对角块$A_{ii}$的特征值集之并$\bigcup_{i=1}^{b}\lambda(A_{ii})$(见问题4.1). 我们计算的典范型将是块三角阵，并且通过分裂大的对角块为较小的块进行计算. 若用一个复矩阵A开始则最终的对角块将是1×1阶的，因而最终的典范型将为三角阵. 若用一个实矩阵A开始，则最终的典范型将有1×1阶对角块(对应于实特征值)和2×2阶对角块(对应于一对复共轭特征值)；这样的一个块三角阵称为准三角阵(quasi-triangular).

求出(块)三角阵的特征向量也容易. 见4.2.1节.

定义4.3　设S是任意非奇异阵. 则称A和$B=S^{-1}AS$为相似矩阵，而S是一个相似变换.

命题4.1　设$B=S^{-1}AS$，故A和B相似. 则A和B有相同的特征值，且x(或y)是A的一个右(或左)特征向量当且仅当$S^{-1}x$(或S^*y)是B的一个右(或左)特征向量.

证明　对任意的方阵X和Y利用事实$\det(X\cdot Y)=\det(X)\cdot\det(Y)$，可写$\det(A-\lambda I)=\det(S^{-1}(A-\lambda I)S)=\det(B-\lambda I)$，故$A$和$B$有相同的特征多项式. $Ax=\lambda x$成立当且仅当$S^{-1}ASS^{-1}x=\lambda S^{-1}x$或$B(S^{-1}x)=\lambda(S^{-1}x)$. 类似地，$y^*A=\lambda y^*$成立当且仅当$y^*SS^{-1}AS=\lambda y^*S$或$(S^*y)^*B=\lambda(S^*y)^*$. □

定理 4.1 若尔当典范型. 给定 A, 存在一个非奇异阵 S 使得 $S^{-1}AS=J$, 其中 J 是若尔当典范型. 这意味着 J 是块对角阵, $J=\mathrm{diag}(J_{n_1}(\lambda_1),J_{n_2}(\lambda_2),\cdots,J_{n_k}(\lambda_k))$ 而

$$J_{n_i}(\lambda_i)=\begin{bmatrix}\lambda_i & 1 & & 0\\ & \ddots & \ddots & \\ & & \ddots & 1\\ 0 & & & \lambda_i\end{bmatrix}^{n_i\times n_i}$$

J 不计它的对角块的置换是唯一的.

本定理的证明见有关的线性代数的著作, 例如[110]或[139].

每个 $J_m(\lambda)$ 称为有代数重数(algebraic multiplicity) m 的特征值 λ 的若尔当块. 若某个 $n_i=1$ 而 λ_i 是一个若尔当块仅有的特征值(simple eigenvalue), 则 A 称为可对角化的; 否则它称为亏损(defective)的. 一个 $n\times n$ 阶亏损阵没有 n 个特征向量, 更详尽的如下列命题中所述. 虽然在一定的界限分明意义下亏损阵是"罕见的", 但某些矩阵没有 n 个特征向量的事实是任何设计计算特征值和特征向量算法的人所面临的一个基本的事实. 在 4.3 节中将看到这类矩阵产生的某些困难. 在第 5 章中讨论的对称阵决不会亏损.

141

命题 4.2 一个若尔当块有一个右特征向量 $e_1=[1,0,\cdots,0]^{\mathrm{T}}$, 和一个左特征向量 $e_n=[0,\cdots,0,1]^{\mathrm{T}}$. 所以一个矩阵有匹配其 n 个特征值的 n 个特征向量当且仅当它是可对角化的. 此时, $S^{-1}AS=\mathrm{diag}(\lambda_i)$. 这等价于 $AS=S\mathrm{diag}(\lambda_i)$, 故 S 的第 i 列是关于 λ_i 的一个右特征向量. 这也等价于 $S^{-1}A=\mathrm{diag}(\lambda_i)S^{-1}$, 故 S^{-1} 的第 i 行的共轭转置是关于 λ_i 的一个左特征向量. 若矩阵 A 的所有 n 个特征值是各不相同的, 则 A 是可对角化的.

证明 为简化记号起见设 $J=J_m(\lambda)$. 容易看出 $Je_1=\lambda e_1$ 和 $e_n^{\mathrm{T}}J=\lambda e_n^{\mathrm{T}}$, 故 e_1 和 e_n 分别是 J 的右和左特征向量. 为了看出 J 只有一个右特征向量(直到标量倍), 注意到任何特征向量 x 必须满足 $(J-\lambda I)x=0$, 故 x 在

$$J-\lambda I=\begin{bmatrix}0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1\\ & & & 0\end{bmatrix}.$$

的零空间内. 但 $J-\lambda I$ 的零空间显然为 $\mathrm{span}(e_1)$, 所以只有一个特征向量. 若 A 的所有特征值是各不相同的, 则所有它的若尔当块必须是 1×1 阶的, 故 $J=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)$ 是对角阵. □

例 4.1 用一个机械振动(mechanical vibration)问题来说明特征值和特征向量概念. 我们将看到亏损矩阵产生于自然界的物质的先后关系. 考察图 4-1 中的阻尼质点弹簧系统, 我们将利用它来说明各种各样的特征值问题.

142

牛顿定律 $F=ma$ 应用于这个系统得到

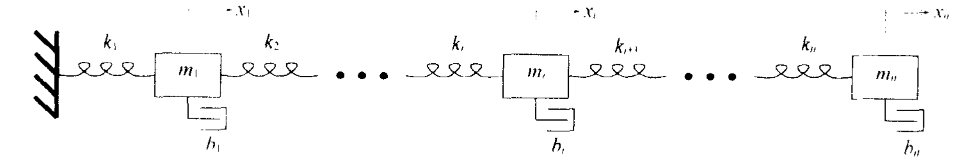
$$\begin{aligned} m_i\ddot{x}_i(t) &= k_i(x_{i-1}(t)-x_i(t)) && \text{从弹簧 } i \text{ 作用到质点 } i \text{ 上的力}\\ &\quad + k_{i+1}(x_{i+1}(t)-x_i(t)) && \text{从弹簧 } i+1 \text{ 作用到质点 } i \text{ 上的力}\\ &\quad - b_i\dot{x}_i(t) && \text{从阻尼器 } i \text{ 作用到质点 } i \text{ 上的力} \end{aligned} \tag{4.2}$$

或

$$M\ddot{x}(t) = -B\dot{x}(t) - Kx(t), \tag{4.3}$$

其中 $M = \mathrm{diag}(m_1, \cdots, m_n)$，$B = \mathrm{diag}(b_1, \cdots, b_n)$，以及

$$\boldsymbol{K} = \begin{bmatrix} k_1+k_2 & -k_2 & & & \\ -k_2 & k_2+k_3 & -k_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-1} & k_{n-1}+k_n & -k_n \\ & & & -k_n & k_n \end{bmatrix}.$$



x_i = 第 i 个质点的位置(0 = 平衡)
m_i = 第 i 个质点的质量
k_i = 第 i 个弹簧的弹力常数
b_i = 第 i 个阻尼器的阻尼常数

图 4-1　阻尼、振动的质点 - 弹簧系统

我们假定所有的质量 m_i 是正的. M 称为质量矩阵，B 是阻尼矩阵(damping matrix)而 K 是刚度矩阵.

电机工程师应用基尔霍夫等定律代替牛顿定律分析线性电路得到类似的方程. 此时 x 代表支线电流，M 代表电感，B 代表电阻而 K 代表导纳(电容的倒数).

利用标准的诀窍把这个二阶微分方程改变为一个一阶微分方程，把变量改变为

$$y(t) = \begin{bmatrix} \dot{x}(t) \\ x(t) \end{bmatrix}.$$

这就得到

$$\dot{y}(t) = \begin{bmatrix} \ddot{x}(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} -M^{-1}B\dot{x}(t) - M^{-1}Kx(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} -\boldsymbol{M}^{-1}\boldsymbol{B} & -\boldsymbol{M}^{-1}\boldsymbol{K} \\ \boldsymbol{I} & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{x}(t) \\ x(t) \end{bmatrix}$$

$$= \begin{bmatrix} -\boldsymbol{M}^{-1}\boldsymbol{B} & -\boldsymbol{M}^{-1}\boldsymbol{K} \\ \boldsymbol{I} & 0 \end{bmatrix} \cdot y(t) \equiv \boldsymbol{A}y(t). \tag{4.4}$$

为求解 $\dot{y}(t)=Ay(t)$,假定 $y(0)$是给定的[即初始位置 $x(0)$和速度 $\dot{x}(0)$是给定的].

写出这个微分方程解的一种方式是 $y(t)=e^{At}y(0)$，其中 e^{At}是矩阵指数. 在 A 是可对角化的特殊情况将给出另一个更初等的解；这对几乎所有的 m_i，k_i 和 b_i 的选择是成立的. 后面将回来考察其他情况.（计算像 e^{At}那样的矩阵函数的一般问题在4.5.1 节和问题4.4 中进一步讨论.） 143

当 A 可对角化时可记 $A=S\Lambda S^{-1}$,其中 $\Lambda=\mathrm{diag}(\lambda,\cdots,\lambda_n)$. 于是 $\dot{y}(t)=Ay(t)$等价于 $\dot{y}(t)=S\Lambda S^{-1}y(t)$或 $S^{-1}\dot{y}(t)=\Lambda S^{-1}y(t)$或 $\dot{z}(t)=\Lambda z(t)$,其中 $z(t)\equiv S^{-1}y(t)$. 这个对角微分方程组 $\dot{z}_i(t)=\lambda_i z_i(t)$有解 $z_i(t)=e^{\lambda_i t}z_i(0)$，故 $y(t)=S\mathrm{diag}(e^{\lambda_1 t},\cdots,e^{\lambda_n t})S^{-1}y(0)=Se^{\Lambda t}S^{-1}y(0)$. 图4-2(见彩插)是4 个质量和弹簧数值解的一个实例.

为观察一个质点 - 弹簧系统中 A 的不可对角化的物理意义，考虑一个单个质量、弹簧和阻尼器的情况，其微分方程可简化为 $m\ddot{x}(t)=-b\dot{x}(t)-kx(t)$，因而 $A=\begin{bmatrix}-b/m & -k/m\\ 1 & 0\end{bmatrix}$. A 的两个特征值为 $\lambda_{\pm}=\frac{b}{2m}(-1\pm(1-\frac{4km}{b^2})^{1/2})$. 当$\frac{4km}{b^2}<1$ 时，系统是超阻尼(overdamped)的，有两个负的实特征值，其平均值是 $-\frac{b}{2m}$. 此时解最终单调地衰减到零. 当$\frac{4km}{b^2}>1$ 时，系统是下阻尼(underdamped)的，有两个实部为 $-\frac{b}{2m}$的复共轭特征值. 此时解当衰减到零时振荡. 这两种情况方程组都是可对角化的，因为特征值是各不相同的. 当$\frac{4km}{b^2}=1$ 时，系统是临界阻尼(critically damped)的. 有两个等于 $-\frac{b}{2m}$的实特征值，而 A 具有这个特征值的单个 2×2 若尔当块. 换言之，不可对角化矩阵构成两种物理状态振荡和单调衰减之间的“边界”. 144

当 A 可对角化但 S 是一个病态矩阵时，致使 S^{-1}很难精确地计算，显式解 $y(t)=Se^{\Lambda t}S^{-1}y(0)$将很不精确，并且数值上是无效的. 我们将不断地利用这个机械系统的例子，因为它说明了相当多的特征问题. ◇

为继续讨论典范型，定义下列特征向量的推广是方便的.

定义4.4 A 的不变子空间是$\mathbb{R}^n$的一个子空间 $\mathbf{X}$，具有性质 $x\in\mathbf{X}$ 蕴含 $Ax\in\mathbf{X}$. 也把这个记为 $A\mathbf{X}\subseteq\mathbf{X}$.

最简单的一维不变子空间是一个特征向量 x 的所有标量倍的集合 span(x). 下面是构造一个较大维数不变子空间类似的方法. 设 $X=[x_1,\cdots,x_m]$，其中 $x_1,\cdots,x_m$ 是任意一组关于 $\lambda_1,\cdots,\lambda_m$ 的线性无关的特征向量. 则 $\mathbf{X}=\mathrm{span}(X)$是一个不变子空间，因为 $x\in\mathbf{X}$ 蕴含对某些标量 α_i,$x=\sum_{i=1}^m\alpha_i x_i$,故 $Ax=\sum_{i=1}^m\alpha_i Ax_i=\sum_{i=1}^m\alpha_i\lambda_i x_i\in\mathbf{X}$. 除非某些特征值 λ_i 等于0，$A\mathbf{X}$ 将等于 $\mathbf{X}$. 下面的命题推广这种情况.

命题4.3 设 A 是 $n\times n$ 阶矩阵，$X=[x_1,\cdots,x_m]$是具有无关列的任意 $n\times m$ 阶矩阵，

并设 $\mathbf{X}=\text{span}(X)$ 是 X 的列张成的 m 维空间. 则 $\mathbf{X}$ 是一个不变子空间当且仅当存在一个 $m\times m$ 阶矩阵 B 使得 $AX=XB$. 此时 B 的 m 个特征值也是 A 的特征值. (当 $m=1$ 时 $X=[x_1]$ 是一个特征向量而 B 是一个特征值).

证明　首先假定 $\mathbf{X}$ 是不变的. 则每个 Ax_i 也在 $\mathbf{X}$ 中, 故每个 Ax_i 必是 X 的一个基底的线性组合, 譬如说, $Ax_i=\sum_{j=1}^{m}x_jb_{ji}$. 这个最后的等式等价于 $AX=XB$. 反之, $AX=XB$ 意味着每个 Ax_i 是 X 的列的线性组合, 故 $\mathbf{X}$ 是不变的.

现设 $AX=XB$. 选择任意的 $n\times(n-m)$ 阶矩阵 $\tilde{X}$ 使得 $\hat{X}=[X,\tilde{X}]$ 非奇异, 则 A 和 $\hat{X}^{-1}A\hat{X}$ 相似, 故有相同的特征值. 记 $\hat{X}^{-1}=\begin{bmatrix}Y^{m\times n}\\ \tilde{Y}^{(n-m)\times n}\end{bmatrix}$, 故 $\hat{X}^{-1}\hat{X}=I$ 推出 $YX=I$ 和 $\tilde{Y}X=0$. 于是 $\hat{X}^{-1}A\hat{X}=\begin{bmatrix}Y\\ \tilde{Y}\end{bmatrix}\cdot[AX,A\tilde{X}]=\begin{bmatrix}YAX & YA\tilde{X}\\ \tilde{Y}AX & \tilde{Y}A\tilde{X}\end{bmatrix}=\begin{bmatrix}YXB & YA\tilde{X}\\ \tilde{Y}XB & \tilde{Y}A\tilde{X}\end{bmatrix}=\begin{bmatrix}B & YA\tilde{X}\\ 0 & \tilde{Y}A\tilde{X}\end{bmatrix}$. 于是由问题4.1知 A 的特征值是 B 的特征值和 $\tilde{Y}A\tilde{X}$ 的特征值之并. □

[145] 例如, 记若尔当典范型 $S^{-1}AS=J=\text{diag}(J_{n_i}(\lambda_i))$ 为 $AS=SJ$, 其中 $S=[S_1,S_2,\cdots,S_k]$ 且 S_i 有 n_i 列(和 $J_{n_i}(\lambda_i)$ 一样, 记号见定理4.1). 于是 $AS=SJ$ 推得 $AS_i=S_iJ_{n_i}(\lambda_i)$, 即 $\text{span}(S_i)$ 是一个不变子空间.

若尔当型告诉我们可能想知道的有关一个矩阵及其特征值、特征向量和不变子空间的每件事情. 也有基于若尔当型的显式公式去计算 e^A 或任何其他的矩阵函数 (见4.5.1节). 但是由于两个数值的理由, 故它对计算若尔当型是坏的公式:

第一个理由: 它是 A 的一个不连续函数, 故任何舍入误差能完全地改变它.

例4.2　设

$$J_n(0)=\begin{bmatrix}0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1\\ & & & 0\end{bmatrix}$$

是一个若尔当型. 对任意小的 ε, 和把 $i\cdot\varepsilon$ 加到 (i,i) 元素上后使特征值改变为 n 个各不相同的值 $i\cdot\varepsilon$, 从而把若尔当型从 $J_n(0)$ 改变为 $\text{diag}(\varepsilon,2\varepsilon,\cdots,n\varepsilon)$. ◇

第二个理由: 一般不能够稳定地计算它. 换言之, 当我们完成计算 S 和 J 后, 不能保证对某个小的 δA 有 $S^{-1}(A+\delta A)S=J$.

例4.3　假如精确地有 $S^{-1}AS=J$, 其中 S 是非常病态的($\kappa(S)=\|S\|\cdot\|S^{-1}\|$非常

大.）假如我们极其幸运，设法精确地计算 S 和计算 J 只有一个微小的误差 δJ，$\|\delta J\|=O(\varepsilon)\|A\|$. 问向后误差多少大？我们得到 $\delta A=S\delta JS^{-1}$，且能断定 $\|\delta A\|\leq\|S\|\cdot\|\delta J\|\cdot\|S^{-1}\|=O(\varepsilon)\kappa(S)\|A\|$. 于是 $\|\delta A\|$ 可能大大超过 $\varepsilon\|A\|$. 这就妨碍向后稳定性. ◇

故我们不去计算 $S^{-1}AS=J$，其中 S 可能是一个任意的病态阵，而我们将限制 S 为正交阵(故 $\kappa_2(S)=1$)以保证稳定性. 我们不能够得到任何像若尔当型一样简单的典范型，但得到几乎同样有效的某种形式.

定理4.2 舒尔典范型. 给定 A，存在一个酉阵 Q 和一个上三角阵 T 使得 $Q^*AQ=T$. A 的特征值是 T 的对角元.

证明 对 n 用归纳法，当 A 是 1×1 阶矩阵时，结论显然成立. 今设 λ 是任意的特征值而 u 是相应规范特征向量，故 $\|u\|_2=1$. 选择 $\widetilde{U}$ 使 $U=[u,\widetilde{U}]$ 是方的酉阵.（注意即使 A 是实的，λ 和 u 可能是复的.）于是

146

$$U^*\cdot A\cdot U=\begin{bmatrix}u^*\\ \widetilde{U}^*\end{bmatrix}\cdot A\cdot[u,\widetilde{U}]=\begin{bmatrix}u^*Au & u^*A\widetilde{U}\\ \widetilde{U}^*Au & \widetilde{U}^*A\widetilde{U}\end{bmatrix}.$$

如命题4.3证明中一样，可记 $u^*Au=\lambda u^*u=\lambda$，$\widetilde{U}^*Au=\lambda\widetilde{U}^*u=0$ 故 $U^*AU\equiv\begin{bmatrix}\lambda & \tilde{a}_{12}\\ 0 & \widetilde{A}_{22}\end{bmatrix}$. 由归纳法，存在一个酉阵 P，故 $P^*\widetilde{A}_{22}P=\widetilde{T}$ 是上三角阵. 于是

$$U^*AU=\begin{bmatrix}\lambda & \tilde{a}_{12}\\ 0 & P\widetilde{T}P^*\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & P\end{bmatrix}\begin{bmatrix}\lambda & \tilde{a}_{12}P\\ 0 & \widetilde{T}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & P^*\end{bmatrix},\text{ 故}$$

$$\begin{bmatrix}1 & 0\\ 0 & P^*\end{bmatrix}U^*AU\begin{bmatrix}1 & 0\\ 0 & P\end{bmatrix}=\begin{bmatrix}\lambda & \tilde{a}_{12}P\\ 0 & \widetilde{T}\end{bmatrix}=T$$

是上三角阵而 $Q=U\begin{bmatrix}1 & 0\\ 0 & P\end{bmatrix}$ 为要求的酉阵. □

注意舒尔型不唯一，因为特征值可能以任意的次序出现在 T 的对角线上.

这种做法引入复数，即使 A 是实阵. 当 A 是实阵时，我们宁可利用只有实数的典范型，因为它对计算来说代价比较低. 如本节开始时提及的那样，这意味着我们必须牺牲三角形典范型而满足于块三角形典范型.

定理4.3 实舒尔典范型. 若 A 是实阵，存在一个实正交阵 V 使得 $V^TAV=T$ 是准上三角阵. 这意味着 T 是对角线上具有 1×1 阶和 2×2 阶的块上三角阵. 它的特征值

是它的对角块的特征值. 1×1 阶对应于实特征值, 2×2 阶对应于一对复共轭特征值.

证明　如前面一样用归纳法. 设 λ 是一个特征值. 若 λ 是实的, 它有一个实特征向量 $\boldsymbol{u}$, 我们像上面的定理中那样处理. 若 λ 是复的, 设 $\boldsymbol{u}$(必然)是一个复特征向量, 因而 $\boldsymbol{A}\boldsymbol{u} = \lambda \boldsymbol{u}$. 因为 $\overline{\boldsymbol{A}}\,\overline{\boldsymbol{u}} = \boldsymbol{A}\,\overline{\boldsymbol{u}} = \overline{\lambda}\overline{\boldsymbol{u}}$, 所以 $\overline{\lambda}$ 和 $\overline{\boldsymbol{u}}$ 也是一对特征值/特征向量. 设 $\boldsymbol{u}_R = \frac{1}{2}\boldsymbol{u} + \frac{1}{2}\overline{\boldsymbol{u}}$ 是 $\boldsymbol{u}$ 的实部, $\boldsymbol{u}_I = \frac{1}{2\mathrm{i}}\boldsymbol{u} - \frac{1}{2\mathrm{i}}\overline{\boldsymbol{u}}$ 是虚部. 于是 $\mathrm{span}\{\boldsymbol{u}_R, \boldsymbol{u}_I\} = \mathrm{span}\{\boldsymbol{u}, \overline{\boldsymbol{u}}\}$ 是一个二维不变子空间. 设 $\widetilde{\boldsymbol{U}} = [\boldsymbol{u}_R, \boldsymbol{u}_I]$ 而 $\widetilde{\boldsymbol{U}} = \boldsymbol{Q}\boldsymbol{R}$ 是其 QR 分解. 于是 $\mathrm{span}\{\boldsymbol{Q}\} = \mathrm{span}\{\boldsymbol{u}_R, \boldsymbol{u}_I\}$ 是不变的. 选择 $\widetilde{\boldsymbol{Q}}$ 使得 $\boldsymbol{U} = [\boldsymbol{Q}, \widetilde{\boldsymbol{Q}}]$ 是实正交阵, 并计算

$$\boldsymbol{U}^{\mathrm{T}} \cdot \boldsymbol{A} \cdot \boldsymbol{U} = \begin{bmatrix} \boldsymbol{Q}^{\mathrm{T}} \\ \widetilde{\boldsymbol{Q}}^{\mathrm{T}} \end{bmatrix} \cdot \boldsymbol{A} \cdot [\boldsymbol{Q}, \widetilde{\boldsymbol{Q}}] = \begin{bmatrix} \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} & \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{Q}} \\ \widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} & \widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{Q}} \end{bmatrix}.$$

因为 $\boldsymbol{Q}$ 张成一个不变子空间, 所以存在一个 2×2 阶矩阵 $\boldsymbol{B}$ 使得 $\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{B}$. 正如命题4.3 证明中那样, 可以记 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{B} = \boldsymbol{B}$ 且 $\widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = \widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{B} = 0$, 故 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{U} =$

147

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{Q}} \\ 0 & \widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{Q}} \end{bmatrix}.$$ 再对 $\widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{Q}}$ 应用归纳法即可证明. □

由舒尔型计算特征向量

设 $\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{T}$ 是舒尔型. 则当 $\boldsymbol{T}\boldsymbol{x} = \lambda \boldsymbol{x}$ 时, 有 $\boldsymbol{A}\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{T}\boldsymbol{x} = \lambda \boldsymbol{Q}\boldsymbol{x}$, 故 $\boldsymbol{Q}\boldsymbol{x}$ 是 $\boldsymbol{A}$ 的一个特征向量. 因此为求 $\boldsymbol{A}$ 的特征向量, 只要求 $\boldsymbol{T}$ 的特征向量就足够了.

假如 $\lambda = t_{ii}$ 的重数为 1(即它为单根). 记 $(\boldsymbol{T} - \lambda \boldsymbol{I})\boldsymbol{x} = 0$ 为

$$0 = \begin{bmatrix} \boldsymbol{T}_{11} - \lambda \boldsymbol{I} & \boldsymbol{T}_{12} & \boldsymbol{T}_{13} \\ 0 & 0 & \boldsymbol{T}_{23} \\ 0 & 0 & \boldsymbol{T}_{33} - \lambda \boldsymbol{I} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} (\boldsymbol{T}_{11} - \lambda \boldsymbol{I})x_1 + \boldsymbol{T}_{12}x_2 + \boldsymbol{T}_{13}x_3 \\ \boldsymbol{T}_{23}x_3 \\ (\boldsymbol{T}_{33} - \lambda \boldsymbol{I})x_3 \end{bmatrix},$$

其中 $\boldsymbol{T}_{11}$ 是 $(i-1) \times (i-1)$ 阶矩阵, $T_{22} = \lambda$ 是 1×1 阶矩阵, $\boldsymbol{T}_{33}$ 是 $(n-i) \times (n-i)$ 阶矩阵, 它们与 $\boldsymbol{x}$ 的分割相一致. 因为 λ 是单根, $\boldsymbol{T}_{11} - \lambda \boldsymbol{I}$ 和 $\boldsymbol{T}_{33} - \lambda \boldsymbol{I}$ 都是非奇异的, 故 $(\boldsymbol{T}_{33} - \lambda \boldsymbol{I})x_3 = 0$ 推出 $x_3 = 0$. 所以 $(\boldsymbol{T}_{11} - \lambda \boldsymbol{I})x_1 = -\boldsymbol{T}_{12}x_2$. 选择(任意的) $x_2 = 1$ 意味着 $x_1 = -(\boldsymbol{T}_{11} - \lambda \boldsymbol{I})^{-1}\boldsymbol{T}_{12}$, 故

$$\boldsymbol{x} = \begin{bmatrix} (\lambda \boldsymbol{I} - \boldsymbol{T}_{11})^{-1}\boldsymbol{T}_{12} \\ 1 \\ 0 \end{bmatrix}.$$

换言之, 我们只需求解一个关于 x_1 的三角形方程组. 为了从实舒尔型求一个实特

征向量，我们需要求解一个准三角形方程组. 从实舒尔型出发只利用实数运算计算复的特征向量也只是需要方程求解，但是有点儿复杂. 细节见 LAPACK 中的子程序 strevc.

4.3 扰动理论

本节中将集中于了解何时特征值是病态的，因而很难精确地计算. 除了提供计算的特征值的误差界外，也将把特征值条件数与有关的量，包括到具有无限地(infinitely)病态特征值最近的矩阵的距离，以及特征向量矩阵的条件数联系起来.

首先研究何时特征值有无限的条件数. 这是如同下例说明的重特征值情况.

例 4.4 设

$$A=\begin{bmatrix}0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1\\ \epsilon & & & 0\end{bmatrix}$$

148

是一个 $n\times n$ 阶矩阵. 则 A 有特征多项式 $\lambda^n-\epsilon=0$，故 $\lambda=\sqrt[n]{\epsilon}$($n$ 个可能的值). ϵ 的 n 次根对小的 ϵ 增长大大地快于任何 ϵ 的倍数. 更正式地，条件数是无限的，因为对 $n\geq 2$ 在 $\epsilon=0$ 处 $\frac{d\lambda}{d\epsilon}=\frac{\epsilon^{\frac{1}{n}-1}}{n}=\infty$. 例如，取 $n=16$ 和 $\epsilon=10^{-16}$，则每个特征值 $|\lambda|=0.1$. ◇

故当一个特征值是"接近于重特征值"时，我们预料一个大的条件数，即存在一个小的 δA 使得 $A+\delta A$ 确实有重特征值. 然而，具有一个无限的条件数并不意味着它们不能够算出具有正确的数位.

命题 4.4 *A 的特征值是 A 的连续函数，即使它们是不可微的.*

证明 只要证明多项式之根的连续性就足够了，因为特征多项式的系数是矩阵元素的连续(多项式)函数. 利用来自复分析的幅角原理[2]：多项式 p 位于简单闭曲线 γ 内部之根的个数是 $\frac{1}{2\pi i}\oint_\gamma\frac{p'(z)}{p(z)}dz$. 若 p 只改变一点点，则 $\frac{p'(z)}{p(z)}$ 只改变一点点，因而 $\frac{1}{2\pi i}\oint_\gamma\frac{p'(z)}{p(z)}$ 只改变一点点. 但因为它是一个整数，所以它必定是常数，故位于曲线 γ 内部之根的个数是常数. 这意味着这些根不可能经过曲线 γ 的外部(不管 γ 是如何小，假定我们用很小的量扰动 p)，所以根一定是连续的. □

下面将集中讨论计算单重特征值的条件数. 若 λ 是 A 的单重特征值且 δA 是小的，则可以认为 $A+\delta A$ 的一个特征值 $\lambda+\delta\lambda$ "对应于" λ. 它是最接近于 λ 的值. 可以容易地计算一个单重特征值的条件数.

定理 4.4 *设 λ 是 A 具有右特征向量 x 和左特征向量 y 的单重特征值，规范化使得 $\|x\|_2=\|y\|_2=1$. 设 $\lambda+\delta\lambda$ 是 $A+\delta A$ 对应的特征值. 则*

$$\delta\lambda=\frac{y^*\delta Ax}{y^*x}+O(\|\delta A\|^2)$$

或

$$|\delta\lambda| \leq \frac{\|\delta A\|}{|y^*x|} + O(\|\delta A\|^2) = \sec\Theta(y,x)\|\delta A\| + O(\|\delta A\|^2),$$

其中 $\Theta(y,x)$ 是 y 和 x 之间的锐角. 换言之, $\sec\Theta(y,x) = 1/|y^*x|$ 是特征值 λ 的条件数.

证明　从 $(A+\delta A)(x+\delta x) = (\lambda+\delta\lambda)(x+\delta x)$ 减去 $Ax = \lambda x$ 得到

$$A\delta x + \delta Ax + \delta A\delta x = \lambda\delta x + \delta\lambda x + \delta\lambda\delta x.$$

不计二阶项(那些具有两个"δ 项"作为因子的: $\delta A\delta x$ 和 $\delta\lambda\delta x$)并用 y^* 相乘得到

149

$y^*A\delta x + y^*\delta Ax = y^*\lambda\delta x + y^*\delta\lambda x$.

因为 $y^*A\delta x$ 消去 $y^*\lambda\delta x$, 所以如要求的那样可以解得 $\delta\lambda = (y^*\delta Ax)/(y^*x)$. □

注意一个若尔当块分别有右和左特征向量 e_1 和 e_n, 故它的特征值的条件数是 $1/|e_n^*e_1| = 1/0 = \infty$, 这与我们前面的分析一致.

在另一极端, 在对称矩阵的重要特殊情况中, 条件数是1, 所以特征值总由数据精确地确定.

推论 4.1　设 A 对称(或正规: $AA^* = A^*A$). 则 $|\delta\lambda| \leq \|\delta A\| + O(\|\delta A\|^2)$.

证明　若 A 对称或正规, 则其特征向量都是正交的. 即 $Q^*AQ = \Lambda, QQ^* = I$. 故右特征向量 x(Q 的列)和左特征向量 y(Q^* 的行的共轭转置)相等, 且 $1/|y^*x| = 1$. □

为观察各式各样的数值例子, 运行问题4.14中提到的Matlab代码.

后面在定理5.1中将证明若 $\delta A = \delta A^T$, 则不管 $\|\delta A\|_2$ 如何大, 事实上都有 $|\delta\lambda| \leq \|\delta A\|_2$.

定理4.4仅对充分小的 $\|\delta A\|$ 有用. 我们可以把 $O(\|\delta A\|^2)$ 项删掉, 并且以条件数增加 n 倍的代价, 得到一个对任何大小扰动 $\|\delta A\|$ 成立的简单的定理.

定理 4.5　Bauer-Fike.　设 A 全部都是单重特征值(即 A 可对角化). 称这些特征值为 λ_i 它具有右和左特征向量 x_i 和 y_i, 规范化后 $\|x_i\|_2 = \|y_i\|_2 = 1$. 则 $A+\delta A$ 的特征值位于圆盘 B_i 中, 其中 B_i 的中心为 λ_i, 半径为 $n\frac{\|\delta A\|_2}{|y_i^*x_i|}$.

证明将利用下面重复列出的Gershgorin定理(定理2.9).

Gershgorin 定理　设 B 是一个任意的矩阵. 则 B 的特征值 λ 位于由 $|\lambda - b_{ii}| \leq \sum_{j\neq i}|b_{ij}|(i = 1,\cdots,n)$ 所定义的 n 个圆盘的并中.

我们还需要两个简单的引理.

引理 4.1　设 $S = [x_1,\cdots,x_n]$ 是右特征向量的非奇异阵. 则

$$S^{-1} = \begin{bmatrix} y_1^*/y_1^*x_1 \\ y_2^*/y_2^*x_2 \\ \vdots \\ y_n^*/y_n^*x_n \end{bmatrix}.$$

150

引理的证明　我们知道 $AS = S\Lambda$, 其中 $\Lambda = \mathrm{diag}(\lambda_1,\cdots,\lambda_n)$, 因为 S 的列 x_i 是特征向量. 这等价于 $S^{-1}A = \Lambda S^{-1}$, 故 S^{-1} 的行是左特征向量 y_i 的共轭转置. 因此对某些常数 c_i.

$$S^{-1}=\begin{bmatrix} y_1^* \cdot c_1 \\ \vdots \\ y_n^* \cdot c_n \end{bmatrix}$$

但 $I=S^{-1}S$，故 $1=(S^{-1}S)_{ii}=y_i^*x_i\cdot c_i$ 且正如要求的那样 $c_i=\dfrac{1}{y_i^*x_i}$. □

引理 4.2 若(任意矩阵) S 的每一列的 2-范数等于 1，则 $\|S\|_2\leqslant\sqrt{n}$. 类似地，若矩阵的每一行的 2-范数等于 1，则矩阵的 2-范数至多为 $\sqrt{n}$.

引理的证明 $\|S\|_2=\|S^T\|_2=\max_{\|x\|_2=1}\|S^Tx\|_2$. 由柯西－施瓦茨不等式 S^Tx 的每个分量以 1 为界，故 $\|S^Tx\|_2\leqslant\|[1,\cdots,1]^T\|_2=\sqrt{n}$. □

Bauer-Fike 定理的证明 将对 $S^{-1}(A+\delta A)S=\Lambda+F$ 应用 Gershgorin 定理，其中 $\Lambda=S^{-1}AS=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)$，$F=S^{-1}\delta AS$. 想法是证明 $A+\delta A$ 的特征值位于中心在 λ_i 具有给定半径的球中. 为此，我们取包含 $\Lambda+F$ 特征值的圆盘，它们是由 Gershgorin 定理所定义的

$$|\lambda-(\lambda_i+f_{ii})|\leqslant\sum_{j\neq i}|f_{ij}|,$$

稍稍扩大它们得到圆盘

$$\begin{aligned}|\lambda-\lambda_i| &\leqslant \sum_j |f_{ij}| \\ &\leqslant n^{1/2}\cdot\Big(\sum_j |f_{ij}|^2\Big)^{1/2} \quad \text{由柯西－施瓦茨定理} \\ &= n^{1/2}\cdot\|F(i,:)\|_2. \end{aligned} \tag{4.5}$$

现在需要界定 $F=S^{-1}\delta AS$ 的第 i 行 $F(i,:)$ 的 2-范数：

$$\begin{aligned}\|F(i,:)\|_2 &= \|(S^{-1}\delta AS)(i,:)\|_2 \\ &\leqslant \|(S^{-1})(i,:)\|_2\cdot\|\delta A\|_2\cdot\|S\|_2 \quad \text{由引理 1.7} \\ &\leqslant \frac{n^{1/2}}{|y_i^*x_i|}\cdot\|\delta A\|_2 \quad \text{由引理 4.1 和 4.2}\end{aligned}$$

同(4.5)式相结合就证明了本定理. □ [151]

我们不希望留下这样的印象，以为完全不能算出具有任意精度的重特征值仅仅因为它们有无限的条件数. 我们确实期望得到一些正确的尾数(fraction)而不是失去确定的数位. 为了说明，考虑一个双重特征值 1 的 2×2 阶矩阵：$A=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}$. 若把(2.1)元素(最敏感)从 0 扰动到机器精度 ε，则特征值从 1 改变到 $1\pm\sqrt{\varepsilon}$. 换言之，计算的特征值与真正的特征值一半精度相符. 更一般地，对三重根，我们预期得到正确数字的 1/3 左右，对更高重根等的讨论，见问题 1.20.

现在来讨论其他问题中也存在的条件数的几何性质. 记得矩阵求逆条件数 $\|A\|\cdot$

$\|A^{-1}\|$的性质：它的倒数度量到最接近的奇异矩阵，即具有无限条件数的矩阵的距离（见定理2.1）．关于特征值一个类似的事实成立．因为重特征值有无限条件数，所以具有重特征值的矩阵集合对计算特征值来说像奇异矩阵对矩阵求逆那样扮演相同的角色，其中“接近奇异”蕴含病态．

定理 4.6　设 λ 是 A 的单重特征值，具有单位右和左特征向量 x 和 y 且条件数 $c=1/|y^*x|$．则存在一个 δA 使得 $A+\delta A$ 有重特征值 λ，且

$$\frac{\|\delta A\|_2}{\|A\|_2}\leqslant\frac{1}{\sqrt{c^2-1}}.$$

当 $c\gg 1$，即特征值病态时，距离的上界是 $1/\sqrt{c^2-1}\approx 1/c$，即它是条件数的倒数．

证明　首先，不失一般性可以假定 A 为具有 $a_{11}=\lambda$ 的上三角阵(舒尔型)．这是因为取 A 为舒尔型等价于用 $T=Q^*AQ$ 代替 A，其中 Q 是酉阵．若 x 和 y 是 A 的(右和左)特征向量，则 Q^*x 和 Q^*y 是 T 的(右和左)特征向量．因为 $(Q^*y)^*(Q^*x)=y^*QQ^*x=y^*x$，所以改变成舒尔型不会改变 λ 的条件数．[另一种说法是条件数是 x 和 y 之间夹角 $\Theta(x,y)$ 的正割，而把 x 改变为 Q^*x 和把 y 改变为 Q^*y 正好以同样的方式旋转 x 和 y 而不改变它们之间的夹角．]

不失一般性，可以假定 $A=\begin{bmatrix}\lambda & A_{12}\\ 0 & A_{22}\end{bmatrix}$．则 $x=e_1$ 而 y 平行于 $\tilde{y}=[1,A_{12}(\lambda I-A_{22})^{-1}]^*$，或 $y=\tilde{y}/\|\tilde{y}\|_2$．于是

$$c=\frac{1}{|y^*x|}=\frac{\|\tilde{y}\|_2}{\|\tilde{y}^*x\|}=\|\tilde{y}\|_2=(1+\|A_{12}(\lambda I-A_{22})^{-1}\|_2^2)^{1/2}$$

或

[152]

$$\sqrt{c^2-1}=\|A_{12}(\lambda I-A_{22})^{-1}\|_2\leqslant\|A_{12}\|_2\cdot\|(\lambda I-A_{22})^{-1}\|_2\leqslant\frac{\|A\|_2}{\sigma_{\min}(\lambda I-A_{22})}.$$

由最小奇异值的定义，存在一个 δA_{22}，其中 $\|\delta A_{22}\|_2=\sigma_{\min}(\lambda I-A_{22})$ 使得 $A_{22}+\delta A_{22}-\lambda I$ 是奇异的；即 λ 是 $A_{22}+\delta A_{22}$ 的特征值．于是 $\begin{bmatrix}\lambda & A_{12}\\ 0 & A_{22}+\delta A_{22}\end{bmatrix}$ 在 λ 有双重特征值，正如所要求的那样，其中

$$\|\delta A_{22}\|_2=\sigma_{\min}(\lambda I-A_{22})\leqslant\frac{\|A\|_2}{\sqrt{c^2-1}}.$$ □

最后，我们把特征值的条件数与对角化 A 的任何相似矩阵 S 的最小可能条件数 $\|S\|\cdot\|S^{-1}\|$ 联系起来：$S^{-1}AS=\Lambda=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)$．定理说明如果任何特征值有一个大的条件数，则 S 必有一个差不多相等的大条件数．换言之，求(最差的)特征值的条件数与约化矩阵为对角型的条件数是几乎相同的．

定理 4.7 设 A 是可对角化的，具有特征值 λ_i 和分别具有右和左特征向量 x_i 和 y_i，规范化后使得 $\|x_i\|_2=\|y_i\|_2=1$. 假定 S 满足 $S^{-1}AS=\Lambda=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)$. 则 $\|S\|_2\cdot\|S^{-1}\|_2\geqslant\max_i 1/|y_i^*x_i|$. 若选择 $S=[x_1,\cdots,x_n]$，则 $\|S\|_2\cdot\|S^{-1}\|_2\leqslant n\cdot\max_i 1/|y_i^*x_i|$，即 S 的条件数不超过它的最小值的 n 倍.

证明见[69].

关于特征问题，包括特征向量，不变子空间及对应于一个不变子空间的特征值的条件数的一个综述见 LAPACK 手册的第4章[10]以及[161,237]. 计算这些条件数的算法可利用 LAPACK 的子程序 strsna 和 strsen 或者调用驱动程序 sgeevx 和 sgeesx.

4.4 非对称特征问题的算法

我们以较简单的一些情况开始建立最终的算法——位移的 QR 算法. 为简单起见，假定 A 是实阵.

第一个且最简单的算法是幂法(4.4.1 节)，它只能求 A 绝对值最大的特征值以及相应的特征向量. 为求其他的特征值和特征向量，我们就某个位移 σ 对 $(A-\sigma I)^{-1}$ 应用幂法，该算法称为逆迭代(4.4.2 节)；注意 $(A-\sigma I)^{-1}$ 的最大特征值是 $1/(\lambda_i-\sigma)$，其中 λ_i 是最接近于 σ 的特征值，故可以通过选择 σ 来选择要求的特征值. 紧接着对幂法的改进让我们同时计算整个不变子空间而不只是单个特征向量；[153] 这个算法称为正交迭代(4.4.3 节). 最后，重组正交迭代使之方便地应用于 $(A-\sigma I)^{-1}$ 而不是 A；这个算法称为 QR 迭代(4.4.4 节).

数学上讲，QR 迭代(带位移 σ)是最终的算法. 但是为了使它实际使用足够地快速和可靠(4.4.5 节)，有好几个问题留待解决. 4.4.6 节讨论使 QR 迭代快速所设计的第一个变换：把 A 从稠密阵约化成上海森伯格(Hessenberg)型（非零元仅在主对角线上和次对角线以上，包括第一次对角线）. 随后的几节描述如何对上海森伯格阵(upper Hessenberg form)有效地执行 QR 迭代.（4.4.7 节指出在对称特征值问题和 SVD 中如何简化上海森伯格型.）

4.4.1 幂法

算法 4.1 幂法：给定 x_0，作迭代

$i=0$

重复

$y_{i+1}=Ax_i$

$x_{i+1}=y_{i+1}/\|y_{i+1}\|_2$ （近似特征向量）

$\tilde{\lambda}_{i+1}=x_{i+1}^{\mathrm{T}}Ax_{i+1}$ （近似特征值）

$i=i+1$

直到收敛

首先将此算法应用于$A=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)$，$|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|$这种非常简单的情况．此时特征向量正好是单位矩阵的列e_i．注意x_i也能写成$x_i=A^ix_0/\|A^ix_0\|_2$，因为因子$1/\|y_{i+1}\|_2$仅将x_{i+1}调节为一个单位向量而并不改变其方向．于是得到

$$A^ix_0\equiv A^i\begin{bmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{bmatrix}=\begin{bmatrix}\xi_1\lambda_1^i\\ \xi_2\lambda_2^i\\ \vdots\\ \xi_n\lambda_n^i\end{bmatrix}=\xi_1\lambda_1^i\begin{bmatrix}1\\ \frac{\xi_2}{\xi_1}\left(\frac{\lambda_2}{\lambda_1}\right)^i\\ \vdots\\ \frac{\xi_n}{\xi_1}\left(\frac{\lambda_n}{\lambda_1}\right)^i\end{bmatrix},$$

其中我们已设$\xi_1\neq0$．因为所有的分式λ_j/λ_1绝对值小于1，所以A^ix_0变成越来越几乎平行于e_1，故$x_i=A^ix_0/\|A^ix_0\|_2$变成越来越接近于最大特征值λ_1对应的特征向量$\pm e_1$．收敛速度依赖于比值$|\lambda_2/\lambda_1|\geqslant\cdots\geqslant|\lambda_n/\lambda_1|$小于1的程度，比值越小收敛越快．因为$x_i$收敛于$\pm e_1$，所以$\widetilde{\lambda}_i=x_i^TAx_i$收敛于最大特征值$\lambda_1$．

在证明幂法收敛中，我们作了几个假设，最特殊的是A为对角阵．为分析更一般的情况，假设$A=S\Lambda S^{-1}$是可对角化的，$\Lambda=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)$且把特征值整理成$|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|$．记$S=[s_1,\cdots,s_n]$，其中列$s_i$是对应的特征向量且还满足$\|s_i\|_2=1$．在最后的节中有$S=I$．记$x_0=S(S^{-1}x_0)\equiv S([\xi_1,\cdots,\xi_n]^T)$．又因为$A=S\Lambda S^{-1}$，所以可记

154

$$A^i=\underbrace{(S\Lambda S^{-1})\cdots(S\Lambda S^{-1})}_{i\text{ 次}}=S\Lambda^iS^{-1}$$

因为所有的$S^{-1}\cdot S$相乘等于I．由此可得

$$A^ix_0=(S\Lambda^iS^{-1})S\begin{bmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{bmatrix}=S\begin{bmatrix}\xi_1\lambda_1^i\\ \xi_2\lambda_2^i\\ \vdots\\ \xi_n\lambda_n^i\end{bmatrix}=\xi_1\lambda_1^iS\begin{bmatrix}1\\ \frac{\xi_2}{\xi_1}\left(\frac{\lambda_2}{\lambda_1}\right)^i\\ \vdots\\ \frac{\xi_n}{\xi_1}\left(\frac{\lambda_n}{\lambda_1}\right)^i\end{bmatrix}.$$

如前面一样，括号中的向量收敛于e_1，故A^ix_0越来越接近于对应于λ_1的特征向量$Se_1=s_1$的倍数．所以，$\widetilde{\lambda}_i=x_i^TAx_i$收敛于$s_1^TAs_1=s_1^T\lambda_1s_1=\lambda_1$．

本方法有一个小缺点是假定$\xi_1\neq0$，即x_0不是不变子空间$\mathrm{span}\{s_2,\cdots,s_n\}$；若$x_0$是随机选择的话，这是很有可能的．本方法主要的缺点是它仅对绝对值最大的特征值收敛于特征值/特征向量对，并且它的收敛速度依赖于可能接近于1的量$|\lambda_2/\lambda_1|$，于是导致非常慢的收敛．确实，当A是实的且绝对值最大的特征值是复的，有两个最大绝对值$|\lambda_1|=|\lambda_2|$的复共轭特征值，这样上面的分析完全不能工作．在正交矩阵的极端情况，所有的特征值有相同的绝对值1．

为绘制幂法收敛性的图，见 HOMEPAGE/Matlab/powerplot. m.

4.4.2 逆迭代

我们将克服幂法的缺点，仅仅通过把幂法应用于$(\boldsymbol{A}-\sigma\boldsymbol{I})^{-1}$而不是对$\boldsymbol{A}$来加以描述，其中$\sigma$称为一个位移. 这将使迭代收敛于最接近于$\sigma$的特征值而不是$\lambda_1$. 这个方法称为逆迭代或逆幂法.

155

算法 4.2 逆迭代：给定$\boldsymbol{x}_0$，作迭代

$i=0$

重复

$\boldsymbol{y}_{i+1}=(\boldsymbol{A}-\sigma\boldsymbol{I})^{-1}\boldsymbol{x}_i$

$\boldsymbol{x}_{i+1}=\boldsymbol{y}_{i+1}/\|\boldsymbol{y}_{i+1}\|_2$ （近似特征向量）

$\widetilde{\lambda}_{i+1}=\boldsymbol{x}_{i+1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}_{i+1}$ （近似特征值）

$i=i+1$

直到收敛

为分析收敛性，注意$\boldsymbol{A}=\boldsymbol{S}\boldsymbol{\Lambda}\boldsymbol{S}^{-1}$蕴含$\boldsymbol{A}-\sigma\boldsymbol{I}=\boldsymbol{S}(\boldsymbol{\Lambda}-\sigma\boldsymbol{I})\boldsymbol{S}^{-1}$，故$(\boldsymbol{A}-\sigma\boldsymbol{I})^{-1}=\boldsymbol{S}(\boldsymbol{\Lambda}-\sigma\boldsymbol{I})^{-1}\boldsymbol{S}^{-1}$. 于是$(\boldsymbol{A}-\sigma\boldsymbol{I})^{-1}$有像$\boldsymbol{A}$对应于特征值$((\boldsymbol{\Lambda}-\sigma\boldsymbol{I})^{-1})_{jj}=(\lambda_j-\sigma)^{-1}$那样相同的特征向量$\boldsymbol{s}_j$. 如前面一样的分析告诉我们预期$\boldsymbol{x}_i$收敛到绝对值最大的特征值对应的特征向量. 具体地假设$|\lambda_k-\sigma|$小于所有其他的$|\lambda_i-\sigma|$以致$(\lambda_k-\sigma)^{-1}$是绝对值最大的特征值. 如前面一样记$\boldsymbol{x}_0=\boldsymbol{S}[\xi_1,\cdots,\xi_n]^{\mathrm{T}}$，并假设$\xi_k\neq0$. 则

$$(\boldsymbol{A}-\sigma\boldsymbol{I})^{-i}\boldsymbol{x}_0=(\boldsymbol{S}(\boldsymbol{\Lambda}-\sigma\boldsymbol{I})^{-i}\boldsymbol{S}^{-1})\boldsymbol{S}\begin{bmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{bmatrix}=\boldsymbol{S}\begin{bmatrix}\xi_1(\lambda_1-\sigma)^{-i}\\ \vdots\\ \xi_n(\lambda_n-\sigma)^{-i}\end{bmatrix}$$

$$=\xi_k(\lambda_k-\sigma)^{-i}\boldsymbol{S}\begin{bmatrix}\dfrac{\xi_1}{\xi_k}\left(\dfrac{\lambda_k-\sigma}{\lambda_1-\sigma}\right)^i\\ \vdots\\ 1\\ \vdots\\ \dfrac{\xi_n}{\xi_k}\left(\dfrac{\lambda_k-\sigma}{\lambda_n-\sigma}\right)^i\end{bmatrix},$$

其中 1 位于第k个元素. 因为分数$(\lambda_k-\sigma)/(\lambda_i-\sigma)$绝对值小于 1，所以括号中的向量逼近$\boldsymbol{e}_k$，故$(\boldsymbol{A}-\sigma\boldsymbol{I})^{-i}\boldsymbol{x}_0$越来越接近于对应于$\lambda_k$的特征向量$\boldsymbol{S}\boldsymbol{e}_k=\boldsymbol{s}_k$的倍数. 如前

面一样，$\widetilde{\lambda}_i = x_i^T A x_i$ 也收敛于 λ_k.

逆迭代法优于幂法是其具有收敛于任何要求的特征值(最靠近位移 σ 的一个)的能力. 通过选择 σ 非常接近于一个要求的特征值，可以非常快地收敛，因此不像原来的幂法一样受到附近的特征值的接近的限制. 当我们有一个特征值的好的近似值而且只需要它对应的特征向量时，这个方法特别有效(例如，见 5.3.4 节). 后面将说明在不知道特征值时如何选择这样的一个 σ，这就是为什么我们努力首先计算它!

4.4.3 正交迭代

下面的改进将允许我们同时收敛于一个 p-维不变子空间($p>1$)而不是一个特征向量. 它称为正交迭代，有时称为子空间迭代或同时迭代.

156

> **算法 4.3** 正交迭代:设 Z_0 是一个 $n\times p$ 阶正交阵. 则作迭代
>
> $i=0$
>
> 重复
>
> 　　$Y_{i+1} = AZ_i$
>
> 　　分解 $Y_{i+1} = Z_{i+1}R_{i+1}$　　利用算法 3.2(QR 分解)
>
> 　　　　　　　　　　　　　　　　　(Z_{i+1}张成一个近似不变子空间)
>
> $i=i+1$
>
> 直到收敛

下面是本方法的一个正式的分析. 假定 $|\lambda_p| > |\lambda_{p+1}|$. 若 $p=1$,本方法及其分析与幂法一致, 当 $p>1$ 时, 记 $\text{span}\{Z_{i+1}\} = \text{span}\{Y_{i+1}\} = \text{span}\{AZ_i\}$, 故 $\text{span}\{Z_i\} = \text{span}\{A^iZ_0\} = \text{span}\{S\Lambda^iS^{-1}Z_0\}$. 注意

$$
\begin{aligned}
S\Lambda^iS^{-1}Z_0 &= S\,\text{diag}(\lambda_1^i,\cdots,\lambda_n^i)S^{-1}Z_0 \\
&= \lambda_p^i S\begin{bmatrix} (\lambda_1/\lambda_p)^i & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & (\lambda_n/\lambda_p)^i \end{bmatrix} S^{-1}Z_0.
\end{aligned}
$$

因为当 $j\leqslant p$ 时 $\left|\dfrac{\lambda_j}{\lambda_p}\right| \geqslant 1$,而当 $j>p$ 时 $\left|\dfrac{\lambda_j}{\lambda_p}\right| < 1$, 所以得到

$$
\begin{bmatrix} (\lambda_1/\lambda_p)^i & & \\ & \ddots & \\ & & (\lambda_n/\lambda_p)^i \end{bmatrix} S^{-1}Z_0 = \begin{bmatrix} V_i^{p\times p} \\ W_i^{(n-p)\times p} \end{bmatrix},
$$

其中 W_i 像 $(\lambda_{p+1}/\lambda_p)^i$ 一样趋近于零，而 V_i 不趋近于零. 确实，若 V_0 满秩(4.4.1 节

中$\xi_1 \neq 0$假定的推广），则V_i也将满秩. 记特征向量矩阵$S = [s_1,\cdots,s_n] \equiv [S_p^{n\times p},\hat{S}_p^{n\times(n-p)}]$，即$S_p = [s_1,\cdots,s_p]$. 则$S\Lambda^i S^{-1}Z_0 = \lambda_p^i S\begin{bmatrix} V_i \\ W_i \end{bmatrix} = \lambda_p^i(S_p V_i + \hat{S}_p W_i)$. 于是

$$\text{span}(Z_i) = \text{span}(S\Lambda^i S^{-1}Z_0) = \text{span}(S_p V_i + \hat{S}_p W_i) \Rightarrow \text{span}(S_p X_i)$$

收敛于所要求的由前p个特征向量张成的不变子空间$\text{span}(S_p V_i) = \text{span}(S_p)$.

尽管舍入，使用QR分解保持向量张成满秩的$\text{span}\{A^i Z_0\}$. [157]

注意当我们只沿着Z_i的前$\tilde{p} < p$列完成算法的迭代时，如果开始只用Z_0的前$\tilde{p}$列而不是p列. 则它们等同于我们将要计算的列. 换言之，正交迭代对所有的$\tilde{p} = 1, 2,\cdots,p$同时有效地运行算法. 故当所有的特征值具有各不相同的绝对值时，如前面一样的收敛性分析推出对任意的$\tilde{p} \leqslant p$, Z_i的前$\tilde{p} \leqslant p$列收敛于$\text{span}\{s_1,\cdots,s_{\tilde{p}}\}$.

于是在正交迭代算法中可以设$p = n$和$Z_0 = I$. 下面的定理证明在一定的假设下可以利用正交迭代计算A的舒尔型.

定理4.8 考虑对矩阵A用$p = n$和$Z_0 = I$运行正交迭代. 若A的所有特征值具有各不相同的绝对值并且所有的主子阵$S(1:j,1:j)$都满秩，则$A_i \equiv Z_i^T A Z_i$收敛于A的舒尔型，即对角线上有特征值的上三角阵. 特征值会以绝对值递减次序出现.

证明的梗概 对所有的j，关于$S(1:j,1:j)$的非奇异性假设推出X_0非奇异，正如早先的分析所需要的，从几何上看，这意味着在不变子空间$\text{span}\{s_i,\cdots,s_j\}$中没有向量正交于$Z_0 I$的前$j$列张成的空间$\text{span}\{e_1,\cdots,e_j\}$. 首先注意$Z_i$是一个正交方阵，故$A$和$A_i = Z_i^T A Z_i$相似. 记$Z_i = [Z_{1i}, Z_{2i}]$，其中$Z_{1i}$有$p$列，故

$$A_i = Z_i^T A Z_i = \begin{bmatrix} Z_{1i}^T A Z_{1i} & Z_{1i}^T A Z_{2i} \\ Z_{2i}^T A Z_{1i} & Z_{2i}^T A Z_{2i} \end{bmatrix}.$$

因为$\text{span}\{Z_{1i}\}$收敛于A的一个不变子空间，$\text{span}\{AZ_{1i}\}$收敛于同样的子空间，故$Z_{2i}^T AZ_{1i}$收敛于$Z_{2i}^T Z_{1i} = 0$. 因为这对一切$p < n$成立，A_i的每个次对角元收敛于0，故A_i收敛于上三角型，即舒尔型. □

事实上，这个证明指出子阵$Z_{2i}^T AZ_{1i} = A_i(p+1:n,1:p)$应像$|\lambda_{p+1}/\lambda_p|^i$一样将收敛于零. 于是$\lambda_p$应该出现在$A_i$的$(p,p)$元素并像$\max(|\lambda_{p+1}/\lambda_p|^i, |\lambda_p/\lambda_{p-1}|^i)$一样收敛.

例4.5 正交迭代的收敛性态由下列数值实验来说明，其中我们取$\Lambda = \text{diag}(1,2,6,30)$和一个随机阵$S$(条件数约为20)，构成$A = S\cdot\Lambda\cdot S^{-1}$，并用$p = 4$对$A$运行正交迭代19次. 图4-3和图4-4表示算法的收敛性. 图4-3给出计算的特征值中实际的误差$|A_i(p,p) - \lambda_p|$为实线而近似$\max(|\lambda_{p+1}/\lambda_p|^i, |\lambda_p/\lambda_{p-1}|^i)$为虚线. 因为图像在半对数尺度上(基本上)是具有相同斜率的直线，这意味着它们两者都是形如$y = c\cdot r^i$的函数图像，其中c和r是常数，而r(斜率)如上面预测的那样对两者是相同的. [158]

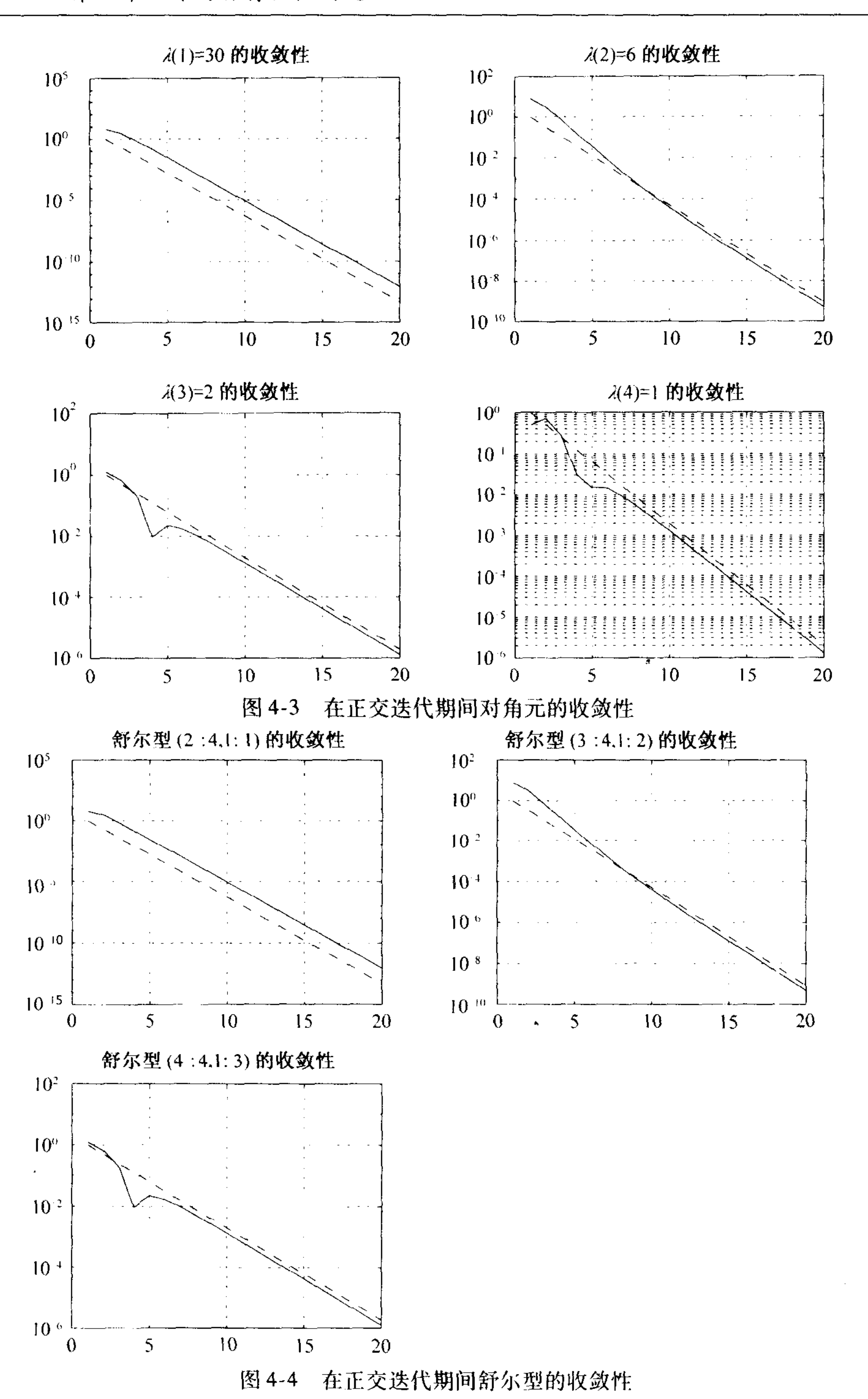


图 4-3 在正交迭代期间对角元的收敛性

图 4-4 在正交迭代期间舒尔型的收敛性

类似地，图4-4标出实际值$\|A_i(p+1:n,1:p)\|_2$为实线而近似$|\lambda_{p+1}/\lambda_p|^i$为虚线；它们也匹配得很好．作为比较下面给出$A_0$和$A_{19}$：

$$A=A_0=\begin{bmatrix}3.5488 & 15.593 & 8.5775 & -4.0123\\ 2.3595 & 24.526 & 14.596 & -5.8157\\ 8.9953\cdot 10^{-2} & 27.599 & 21.483 & -5.8415\\ 1.9227 & 55.667 & 39.717 & -10.558\end{bmatrix},$$

$$A_{19}=\begin{bmatrix}30.000 & -32.557 & -70.844 & 14.984\\ 6.7607\cdot 10^{-13} & 6.0000 & 1.8143 & -0.55754\\ 1.5452\cdot 10^{-23} & 1.1086\cdot 10^{-9} & 2.0000 & -0.25894\\ 7.3360\cdot 10^{-29} & 3.3769\cdot 10^{-15} & 4.9533\cdot 10^{-6} & 1.0000\end{bmatrix}.$$

运行这个例子和类似的例子的Matlab软件是HOMEPAGE/Matlab/qriter. m. ◇

例4.6 为观察为什么定理4.8中关于$S(1:j,1:j)$非奇异性的假设是必要的，假如A是对角阵，在对角线上特征值非递减次序排列．则正交迭代得到$Z_i=\mathrm{diag}(\pm 1)$（具有对角元±1的对角阵）且对一切i，$A_i=A$，故特征值没有移动成递减次序．为观察为什么特征值有各不相同的绝对值的假设是必要的，假如A是正交的，故其所有的特征值绝对值为1．而且算法使A_i基本上保持不变．（行和列可能乘以-1.）◇

4.4.4 QR迭代

下一个目标是为了像4.4.2节一样结合位移和求逆重组正交迭代．这将使它更为有效并去掉特征值大小不同的假设，在定理4.8中证明收敛性需要这个假设．

> **算法4.4** QR迭代：给定A_0，作迭代
> $i=0$
> 重复
> 　　分解$A_i=Q_iR_i$　　（QR分解）
> 　　$A_{i+1}=R_iQ_i$
> 　　$i=i+1$
> 直到收敛

159
≀
160

因为$A_{i+1}=R_iQ_i=Q_i^{\mathrm{T}}(Q_iR_i)Q_i=Q_i^{\mathrm{T}}A_iQ_i$，所以$A_{i+1}$和$A_i$是正交相似的．

我们断言由QR迭代计算的A_i等同于由正交迭代隐式计算矩阵$Z_i^{\mathrm{T}}AZ_i$．

引理4.3 设A_i是由算法4.4计算的矩阵．则$A_i=Z_i^{\mathrm{T}}AZ_i$，其中Z_i是以$Z_0=I$开始从正交迭代（算法4.3）计算的矩阵．因此当所有的特征值有不同的绝对值时A_i收敛于舒尔型．

证明 使用归纳法．假定$A_i=Z_i^{\mathrm{T}}AZ_i$．从算法4.3，我们可以记$AZ_i=Z_{i+1}R_{i+1}$，其中

Z_{i+1}是正交阵而R_{i+1}是上三角阵. 于是$Z_i^{\mathrm{T}}AZ_i = Z_i^{\mathrm{T}}(Z_{i+1}R_{i+1})$是正交阵$Q = Z_i^{\mathrm{T}}Z_{i+1}$和一个上三角阵$R = R_{i+1} = Z_{i+1}^{\mathrm{T}}AZ_i$之积；这一定是QR分解$A_i = QR$，因为QR分解是唯一的（除了可能$Q$的每一列和$R$的每一行乘以$-1$）. 于是

$$Z_{i+1}^{\mathrm{T}}AZ_{i+1} = (Z_{i+1}^{\mathrm{T}}AZ_i)(Z_i^{\mathrm{T}}Z_{i+1}) = R_{i+1}(Z_i^{\mathrm{T}}Z_{i+1}) = RQ.$$

这正好是如何把A_i映射到A_{i+1}的QR迭代，于是如所希望那样$Z_{i+1}^{\mathrm{T}}AZ_{i+1} = A_{i+1}$. □

为观察各种各样说明QR迭代收敛性的数值例子，运行问题4.15中提到的Matlab代码.

从早先的分析，我们知道收敛速度依赖于特征值之比. 为加快收敛，我们使用位移和求逆.

> **算法4.5**　带一个位移的QR迭代：给定A_0，作迭代
>
> $i = 0$
>
> 重复
>
> 　　选择一个接近A的一个特征值的位移σ_i
>
> 　　分解$A_i - \sigma_i I = Q_i R_i$　　（QR分解）
>
> 　　$A_{i+1} = R_i Q_i + \sigma_i I$
>
> 　　$i = i + 1$
>
> 直到收敛

引理4.4　A_i和A_{i+1}正交相似

证明　$A_{i+1} = R_iQ_i + \sigma_i I = Q_i^{\mathrm{T}}Q_iR_iQ_i + \sigma_i Q_i^{\mathrm{T}}Q_i = Q_i^{\mathrm{T}}(Q_iR_i + \sigma_i I)Q_i = Q_i^{\mathrm{T}}A_iQ_i$. □

若R_i非奇异，也可记

[161]　$$A_{i+1} = R_iQ_i + \sigma_i I = R_iQ_iR_iR_i^{-1} + \sigma_i R_iR_i^{-1} = R_i(Q_iR_i + \sigma_i I)R_i^{-1} = R_iA_iR_i^{-1}.$$

若σ_i是A_i的一个精确特征值，则断言QR迭代一步收敛：因为σ_i是一个特征值，$A_i - \sigma_i I$奇异，故R_i奇异，这样R_i的某个对角元必为零. 假如$R_i(n,n) = 0$. 这推出R_iQ_i的最后一行为0，这样$A_{i+1} = R_iQ_i + \sigma_i I$的最后一行等于$\sigma_i e_n^{\mathrm{T}}$,其中$e_n$是$n\times n$阶单位阵的第$n$列. 换言之，除了特征值$\sigma_i$出现在$(n,n)$元素之外$A_{i+1}$的最后一行为零. 这意味着算法已经收敛，因为$A_{i+1}$是具有尾部$1\times 1$块$\sigma_i$的块上三角阵；前$(n-1)\times(n-1)$块$A'$是一个新的较小的特征问题，不再修改$\sigma_i$就可以得到QR迭代：$A_{i+1} = \begin{bmatrix} A' & a \\ 0 & \sigma_i \end{bmatrix}$.

当σ_i不是一个精确特征值时，则当左下块$A_{i+1}(n,1:n-1)$足够小时，我们将认为$A_{i+1}(n,n)$已收敛. 记得从早先的分析我们期待$A_{i+1}(n,1:n-1)$的因子$|\lambda_k - \sigma_i|/\min_{j\neq k}|\lambda_j - \sigma_i|$收缩，其中$|\lambda_k - \sigma_i| = \min_j|\lambda_j - \sigma_i|$. 故若$\sigma_i$是特征值$\lambda_k$的一个非常好的近似时，预期会快速收敛.

下面是另一种方法观察为什么通过重组 QR 迭代隐式地做逆迭代收敛将是快速的. 当 σ_i 是精确的特征值时, $\boldsymbol{Q}_i$ 的最后列 $\underline{\boldsymbol{q}}_i$ 将是 $\boldsymbol{A}_i$ 关于特征值 σ_i 的左特征向量, 因为 $\underline{\boldsymbol{q}}_i^* \boldsymbol{A}_i = \underline{\boldsymbol{q}}_i^* (\boldsymbol{Q}_i \boldsymbol{R}_i + \sigma_i \boldsymbol{I}) = \boldsymbol{e}_n^{\mathrm{T}} \boldsymbol{R}_i + \sigma_i \underline{\boldsymbol{q}}_i^* = \sigma_i \underline{\boldsymbol{q}}_i^*$. 当 σ_i 接近于一个特征值时, 由于下列原因我们预期 $\underline{\boldsymbol{q}}_i$ 接近于一个特征向量: $\underline{\boldsymbol{q}}_i$ 平行于 $((\boldsymbol{A}_i - \sigma_i \boldsymbol{I})^*)^{-1} \boldsymbol{e}_n$ (下面说明为什么). 换言之 $\underline{\boldsymbol{q}}_i$ 与从对 $(\boldsymbol{A}_i - \sigma_i \boldsymbol{I})^*$ 逆迭代所得到的相同(故我们预期它接近于一个左特征向量).

下面是 $\underline{\boldsymbol{q}}_i$ 平行于 $((\boldsymbol{A}_i - \sigma_i \boldsymbol{I})^*)^{-1} \boldsymbol{e}_n$ 的证明. $\boldsymbol{A}_i - \sigma_i \boldsymbol{I} = \boldsymbol{Q}_i \boldsymbol{R}_i$ 推出 $(\boldsymbol{A}_i - \sigma_i \boldsymbol{I}) \boldsymbol{R}_i^{-1} = \boldsymbol{Q}_i$. 两边同时求逆并取共轭转置使得右边的 $\boldsymbol{Q}_i$ 保持不变而把左边变为 $((\boldsymbol{A}_i - \sigma_i \boldsymbol{I})^*)^{-1} \boldsymbol{R}_i^*$, 其最后列为 $((\boldsymbol{A}_i - \sigma_i \boldsymbol{I})^*)^{-1} \cdot [0, \cdots, 0, R_i(n,n)]^{\mathrm{T}}$, 这与 $((\boldsymbol{A}_i - \sigma_i \boldsymbol{I})^*)^{-1}$ 的最后列成比例.

当我们首先尝试计算特征值时, 如何选择 σ_i 是一个准确的近似特征值呢? 关于这个问题在后面将更详细地说明. 但现在注意到几乎收敛于一个实特征值 $\boldsymbol{A}_i(n,n)$ 是接近那个特征值的, 故 $\sigma_i = \boldsymbol{A}_i(n,n)$ 是一个位移的好的选择. 事实上, 它得到局部的二次收敛(quadratic convergence), 这意味着每进行一步正确的数字位数都会加倍. 下面说明为什么会出现二次收敛. 假如在第 i 步 $\|\boldsymbol{A}_i(n,1:n-1)\| / \|\boldsymbol{A}\| \equiv \eta \ll 1$. 若我们置 $\boldsymbol{A}_i(n,1:n-1)$ 精确地为 0, 将使 $\boldsymbol{A}_i$ 为块上三角阵, 故扰动一个真正的特征值 λ_k 使它等于 $\boldsymbol{A}_i(n,n)$. 若这个特征值是远离其他特征值的话, 它将不是病态的, 故这个扰动将是 $O(\eta \|\boldsymbol{A}\|)$. 即, $|\lambda_k - \boldsymbol{A}_i(n,n)| = O(\eta \|\boldsymbol{A}\|)$. 在下次迭代中, 若选择 $\sigma_i = \boldsymbol{A}_i(n,n)$, 我们预期 $\boldsymbol{A}_{i+1}(n,1:n-1)$ 以因子 $|\lambda_k - \sigma_i| / \min_{j \neq k} |\lambda_j - \sigma_i| = O(\eta)$ 收缩, 推出 $\|\boldsymbol{A}_{i+1}(n,1:n-1)\| = O(\eta^2 \|\boldsymbol{A}\|)$ 或 $\|\boldsymbol{A}_{i+1}(n,1:n-1)\| / \|\boldsymbol{A}\| = O(\eta^2)$. 这个方法从 η 到 $O(\eta^2)$ 减少误差是二次收敛. [162]

例 4.7 这里是用例 4.5 中同样的 4×4 阶矩阵 $\boldsymbol{A}_0$ 作为初始的某个位移的 QR 迭代, 位移 $\sigma_i = \boldsymbol{A}_i(4,4)$. 一开始收敛是有点不规则的但靠近终点时最终变成二次, 在最后三步中的每一步 $\|\boldsymbol{A}_i(4,1:3)\| \approx |\boldsymbol{A}_i(4,3)|$ 都是平方近似. 在 $\boldsymbol{A}_i(4,4)$ 中的正确数字位数在最后第 4 步到最后第 2 步上都是倍增的.

$A_0(4,:)=$	$+1.9$	$+56.0$	$+40.0$	-10.558
$A_1(4,:)=$	-0.85	-4.9	$+2.2 \cdot 10^{-2}$	-6.6068
$A_2(4,:)=$	$+0.35$	$+0.86$	$+0.30$	0.74894
$A_3(4,:)=$	$-1.2 \cdot 10^{-2}$	-0.17	-0.70	1.4672
$A_4(4,:)=$	$-1.5 \cdot 10^{-4}$	$-1.8 \cdot 10^{-2}$	-0.38	1.4045
$A_5(4,:)=$	$-3.0 \cdot 10^{-6}$	$-2.2 \cdot 10^{-3}$	-0.50	1.1403
$A_6(4,:)=$	$-1.4 \cdot 10^{-8}$	$-6.3 \cdot 10^{-5}$	$-7.8 \cdot 10^{-2}$	1.0272
$A_7(4,:)=$	$-1.4 \cdot 10^{-11}$	$-3.6 \cdot 10^{-7}$	$-2.3 \cdot 10^{-3}$	0.99941
$A_8(4,:)=$	$+2.8 \cdot 10^{-16}$	$+4.2 \cdot 10^{-11}$	$+1.4 \cdot 10^{-6}$	0.9999996468853453
$A_9(4,:)=$	$-3.4 \cdot 10^{-24}$	$-3.0 \cdot 10^{-18}$	$-4.8 \cdot 10^{-13}$	0.9999999999998767
$A_{10}(4,:)=$	$+1.5 \cdot 10^{-38}$	$+7.4 \cdot 10^{-32}$	$+6.0 \cdot 10^{-26}$	1.000000000000001

在达到 $\boldsymbol{A}_{10}$ 之前, 矩阵的其余部分同样已经朝着收敛前进许多, 故每一步或每二步将很快地算出后面的特征值.

$$A_{10}=\begin{bmatrix} 30.000 & -32.557 & -70.844 & 14.985 \\ 6.1548\cdot 10^{-6} & 6.0000 & 1.8143 & -0.55754 \\ 2.5531\cdot 10^{-13} & 2.0120\cdot 10^{-6} & 2.0000 & -0.25894 \\ 1.4692\cdot 10^{-38} & 7.4289\cdot 10^{-32} & 6.0040\cdot 10^{-26} & 1.0000 \end{bmatrix}.\qquad \diamond$$

4.4.5 使QR迭代有实效

下面是一些必须加以解决的遗留问题，以使算法更有实效：

1. 迭代代价太昂贵. QR分解代价为 $O(n^3)$ flops，故即使我们很幸运每个特征值只做一次迭代，代价将是 $O(n^4)$. 我们寻找总代价仅为 $O(n^3)$ 的算法.

2. 对复特征值我们应如何选择 σ_i 去加速收敛？选择 σ_i 为复数意味着所有的运算必须是复的，增加的代价大约是 A 为实阵时4倍. 当 A 是实阵且收敛于实舒尔型时，我们寻找一个全部利用实数运算的一个算法.

3. 我们如何确认收敛？

[163] 这些问题的解答如下，在后面将更详尽地描述：

1. 首先将把矩阵约化为上海森伯格型；这意味着 A 的第一条次对角线以下为零(即 $a_{ij}=0, i>j+1$)(见4.4.6节)，然后我们应用隐式的(implicity)QR迭代一步，即不计算 Q 或用它明显地相乘(见4.4.8节). 这将把一次QR迭代的代价从 $O(n^3)$ 减少到 $O(n^2)$ 并且如要求的那样全部的代价从 $O(n^4)$ 减少到 $O(n^3)$.
当 A 对称时，将把它约化为上三角型代替上海森伯格型；单个QR迭代的代价进一步减少到 $O(n)$. 这在4.4.7节和第5章中加以讨论.

2. 因为实矩阵的复特征值以复共轭对出现，所以可以同时以 σ_i 和 $\bar{\sigma}_i$ 位移；结果允许我们保持实数运算(见4.4.8节). 若 A 对称，则所有特征值是实的，这不成为一个问题.

3. 当 A_i 的次对角元"足够小"时收敛性出现. 为有助于选择一个实用的阈值，我们利用向后稳定性记号：因为 A_i 是与用正交阵的相似变换 A 有关的，无论如何我们期待 A_i 有大小为 $O(\varepsilon\|A\|)$ 的舍入误差. 所以，数量上比 $O(\varepsilon\|A\|)$ 小的任何 A_i 的次对角元同样可能为零，故我们置它为零[1]. 当 A 为上海森伯格阵，置 $a_{p+1,p}$ 为零将使 A 为块上三角阵 $A=\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22}\end{bmatrix}$，其中 A_{11} 是 $p\times p$ 阶矩阵且 A_{11} 和 A_{22} 都是海森伯格阵. 于是 A_{11} 和 A_{22} 的特征值可独立地求出以得到 A 的特征值. 当所有这些对角块是 1×1 或 2×2 阶矩阵时，算法完成.

---

1. 实际上，我们使用稍为更加严格的条件，用 A 的一个子阵的范数代替 $\|A\|$，考虑到一些矩阵在一个地方大量的元素可能是"坡度的"而其余地方元素是小的. 当两个相邻的次对角线元素之积 $a_{p+1,p}a_{p+2,p+1}$ 足够小时，我们也可置次对角元为零. 细节见LAPACK程序 slahqr.

4.4.6 海森伯格约化

给定实阵 $\boldsymbol{A}$，寻找一个正交的 $\boldsymbol{Q}$ 使 $\boldsymbol{QAQ}^{\mathrm{T}}$ 是上海森伯格阵. 本算法是 QR 分解所用思想的一个简单变形.

例 4.8 用 5×5 的例子来说明海森伯格约化的一般样式. 下面每个 $\boldsymbol{Q}_i$ 都是 5×5 豪斯霍尔德反射，选择 $\boldsymbol{Q}_i$ 使第 i 列中第 $i+2$ 到第 n 个元素为零且保持第 1 个到第 i 个元素不变.

164

1. 选择 $\boldsymbol{Q}_1$ 使

$$\boldsymbol{Q}_1\boldsymbol{A}=\begin{bmatrix} x & x & x & x & x\\ x & x & x & x & x\\ o & x & x & x & x\\ o & x & x & x & x\\ o & x & x & x & x \end{bmatrix}\text{和 } \boldsymbol{A}_1\equiv\boldsymbol{Q}_1\boldsymbol{A}\boldsymbol{Q}_1^{\mathrm{T}}=\begin{bmatrix} x & x & x & x & x\\ x & x & x & x & x\\ o & x & x & x & x\\ o & x & x & x & x\\ o & x & x & x & x \end{bmatrix}.$$

$\boldsymbol{Q}_1$ 保持 $\boldsymbol{Q}_1\boldsymbol{A}$ 的第 1 行不变，而 $\boldsymbol{Q}_1^{\mathrm{T}}$ 保持 $\boldsymbol{Q}_1\boldsymbol{A}\boldsymbol{Q}_1^{\mathrm{T}}$ 的第 1 列包括零元素不变.

2. 选择 $\boldsymbol{Q}_2$ 使

$$\boldsymbol{Q}_2\boldsymbol{A}_1=\begin{bmatrix} x & x & x & x & x\\ x & x & x & x & x\\ o & x & x & x & x\\ o & o & x & x & x\\ o & o & x & x & x \end{bmatrix}\text{和 } \boldsymbol{A}_2\equiv\boldsymbol{Q}_2\boldsymbol{A}_1\boldsymbol{Q}_2^{\mathrm{T}}=\begin{bmatrix} x & x & x & x & x\\ x & x & x & x & x\\ o & x & x & x & x\\ o & o & x & x & x\\ o & o & x & x & x \end{bmatrix}.$$

$\boldsymbol{Q}_2$ 只改变 $\boldsymbol{A}_1$ 最后三行，而 $\boldsymbol{Q}_2^{\mathrm{T}}$ 保持 $\boldsymbol{Q}_2\boldsymbol{A}_1\boldsymbol{Q}_2^{\mathrm{T}}$ 前 2 列包括零元素不变.

3. 选择 $\boldsymbol{Q}_3$ 使

$$\boldsymbol{Q}_3\boldsymbol{A}_2=\begin{bmatrix} x & x & x & x & x\\ x & x & x & x & x\\ o & x & x & x & x\\ o & o & x & x & x\\ o & o & o & x & x \end{bmatrix}\text{和 } \boldsymbol{A}_3=\boldsymbol{Q}_3\boldsymbol{A}_2\boldsymbol{Q}_3^{\mathrm{T}}=\begin{bmatrix} x & x & x & x & x\\ x & x & x & x & x\\ o & x & x & x & x\\ o & o & x & x & x\\ o & o & o & x & x \end{bmatrix}.$$

这是一个上海森伯格阵，合在一起 $\boldsymbol{A}_3=(\boldsymbol{Q}_3\boldsymbol{Q}_2\boldsymbol{Q}_1)\cdot\boldsymbol{A}(\boldsymbol{Q}_3\boldsymbol{Q}_2\boldsymbol{Q}_1)^{\mathrm{T}}\equiv\boldsymbol{QAQ}^{\mathrm{T}}$. ◇

海森伯格约化的一般算法如下：

算法 4.6 约化到上海森伯格型：

若 $\boldsymbol{Q}$ 是要求的，置 $\boldsymbol{Q}=\boldsymbol{I}$

for $i=1:n-2$

  $u_i=\text{House}(\boldsymbol{A}(i+1:n,i))$

  $\boldsymbol{P}_i=\boldsymbol{I}-2\boldsymbol{u}_i\boldsymbol{u}_i^{\mathrm{T}}$ /* $\boldsymbol{Q}_i=\text{diag}(\boldsymbol{I}^{i\times i},\boldsymbol{P}_i)$ */

  $\boldsymbol{A}(i+1:n,i:n)=\boldsymbol{P}_i\cdot\boldsymbol{A}(i+1:n,i:n)$

（续）

$$A(1:n,i+1:n)=A(1:n,i+1:n)\cdot P_i$$

若 Q 是要求的

$$Q(i+1:n,i:n)=P_i\cdot Q(i+1:n,i:n) \quad /^* Q=Q_i\cdot Q\ ^*/$$

end if

[165] end for

正如QR分解那样，不明显地构成 P_i 而代之用 $I-2u_iu_i^T$ 通过矩阵－向量运算相乘．类似于QR分解，向量 u_i 也可存放在次对角线下面．如问题3.17中所述，使用3级BLAS可以应用它们．本算法可以利用Matlab指令hess或LAPACK程序sgehrd来实现．

容易算出浮点运算次数为 $\frac{10}{3}n^3+O(n^2)$ 或当还要计算乘积 $Q=Q_{n-1}\cdots Q_1$ 时为 $\frac{14}{3}n^3+O(n^2)$．

在QR迭代之下海森伯格型的优点是每次迭代只花费 $6n^2+O(n)$ 而不是 $O(n^3)$ 次flops，并且它的形状被保持以致矩阵保持上海森伯格型．

命题4.5 QR迭代保持海森伯格型

证明 容易确认上海森伯格矩阵像 $A_i-\sigma I$ 的QR分解一样得到一个上海森伯格阵 Q（因为 Q 的第 j 列是 $A_i-\sigma I$ 的前 j 列的一个线性组合）．于是容易确认 RQ 保持上海森伯格型，而加上 σI 不改变这个形状． □

定义4.5 上海森伯格阵 H 当所有的次对角元非零时是不可约的．

容易看出若因为 $h_{i+1,i}=0$，H 可约，则它的特征值是它的前 $i\times i$ 阶海森伯格子阵和它的尾部 $(n-i)\times(n-i)$ 阶海森伯格子阵的那些特征值，故我们只需考虑不可约的矩阵．

4.4.7 三对角和双对角约化

若 A 对称，则在海森伯格约化过程中每一步保持 A 对称，故在对称位置上产生零，这意味只需要对半个矩阵操作，约化运算量为 $\frac{4}{3}n^3+O(n^2)$ 或者还要构成 $Q_{n-1}\cdots Q_1$ 时为 $\frac{8}{3}n^3+O(n^2)$．我们称这个算法为三对角约化（tridiagonal reduction）．在第5章中将使用这个算法．这个程序可以利用LAPACK程序ssytrd来实现．

稍为提前一点看看5.4节中计算SVD的讨论，回想3.2.3节对称阵 A^TA 的特征值是 A 的奇异值的平方．最终的SVD算法将利用这个事实，故我们喜欢寻找 A 的一种形式，它蕴含 A^TA 是三对角的．将选择 A 是上双对角的（upper bidiagonal），或非零元仅仅在对角线和第一上对角线上面．因而希望计算正交阵 Q 和 V 使得 QAV 是上

双对角阵. 这个称为上双对角约化(bidiagonal reduction)的算法十分相似于海森伯格和三对角约化算法.

例 4.9 下面是一个说明上双对角约化的一般样式的 4×4 例子.

166

1. 选择 $\boldsymbol{Q}_1$ 使

$$\boldsymbol{Q}_1\boldsymbol{A}=\begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & x & x & x \\ o & x & x & x \end{bmatrix} \text{和 } \boldsymbol{V}_1 \text{ 使 } \boldsymbol{A}_1\equiv\boldsymbol{Q}_1\boldsymbol{A}\boldsymbol{V}_1=\begin{bmatrix} x & x & o & o \\ o & x & x & x \\ o & x & x & x \\ o & x & x & x \end{bmatrix}.$$

$\boldsymbol{Q}_1$ 是一个豪斯霍尔德反射, 而 $\boldsymbol{V}_1$ 是一个保持 $\boldsymbol{Q}_1\boldsymbol{A}$ 第一列不变的豪斯霍尔德反射.

2. 选择 $\boldsymbol{Q}_2$ 使

$$\boldsymbol{Q}_2\boldsymbol{A}_1=\begin{bmatrix} x & x & o & o \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \end{bmatrix} \text{和 } \boldsymbol{V}_2 \text{ 使 } \boldsymbol{A}_2\equiv\boldsymbol{Q}_2\boldsymbol{A}_1\boldsymbol{V}_2=\begin{bmatrix} x & x & o & o \\ o & x & x & o \\ o & o & x & x \\ o & o & x & x \end{bmatrix}.$$

$\boldsymbol{Q}_2$ 是一个保持 $\boldsymbol{A}_1$ 的第一行不变的豪斯霍尔德反射, $\boldsymbol{V}_2$ 是一个保持 $\boldsymbol{Q}_2\boldsymbol{A}_1$ 前二列不变的豪斯霍尔德反射.

3. 选择 $\boldsymbol{Q}_3$ 使

$$\boldsymbol{Q}_3\boldsymbol{A}_2=\begin{bmatrix} x & x & o & o \\ o & x & x & o \\ o & o & x & x \\ o & o & o & x \end{bmatrix} \text{和 } \boldsymbol{V}_3=\boldsymbol{I} \text{ 使 } \boldsymbol{A}_3=\boldsymbol{Q}_3\boldsymbol{A}_2.$$

$\boldsymbol{Q}_3$ 是一个保持 $\boldsymbol{A}_2$ 的前二行不变的豪斯霍尔德反射. ◇

一般地, 若 $\boldsymbol{A}$ 是 $n\times n$ 阶矩阵, 则我们得到正交阵 $\boldsymbol{Q}=\boldsymbol{Q}_{n-1}\cdots\boldsymbol{Q}_1$ 和 $\boldsymbol{V}=\boldsymbol{V}_1\cdots\boldsymbol{V}_{n-2}$ 使得 $\boldsymbol{QAV}=\boldsymbol{A}'$ 是上双对角阵.

注意 $\boldsymbol{A}'^{\mathrm{T}}\boldsymbol{A}'=\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{QAV}=\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{AV}$, 故 $\boldsymbol{A}'^{\mathrm{T}}\boldsymbol{A}'$ 有如 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 一样的特征值, 即 $\boldsymbol{A}'$ 有如 $\boldsymbol{A}$ 一样的奇异值.

这个双对角约化的代价是 $\frac{8}{3}n^3+O(n^2)$ flops, 为计算 $\boldsymbol{Q}$ 和 $\boldsymbol{V}$ 加上另一个 $4n^3+O(n^2)$ flops. 这个程序可以利用 LAPACK 程序 sgebrd 来实现.

4.4.8 隐式位移的 QR 迭代

本节中指出对上海森伯格矩阵如何廉价地执行 QR 迭代. 在下面所述意义上执行过程将是隐式(implicit)的, 我们不明确地计算矩阵 $\boldsymbol{H}$ 的 QR 分解, 而是隐式地构造 $\boldsymbol{Q}$ 为吉文斯旋转和其他简单的正交阵之积. 下面描述的隐式 Q 定理(implicit Q theorem)证明这个隐式构造的 $\boldsymbol{Q}$ 是我们所要的. 然后指出如何结合加速收敛必需的单个位移 σ. 在出现复特征值中为保持实数运算, 我们指出如何做双位移, 即把用复共轭

167

位移σ和$\bar{\sigma}$的两个相继的QR迭代组合起来；这个双位移之后的结果又是实的. 最后，讨论选择位移σ和$\bar{\sigma}$的策略以提供可靠的二次收敛. 然而，最近发现收敛不出现的罕见情况[25,65]，故把寻找完全可靠的且快速的QR迭代执行过程留作一个公开问题.

1. 隐式Q定理

QR迭代最终的执行过程依赖于下列定理.

定理4.9 隐式Q定理. *假如$Q^TAQ=H$是不可约的上海森伯格阵. 则Q的第2列到第n列由Q的第1列唯一地确定(不计符号).*

本定理蕴含在QR算法中从A_i计算$A_{i+1}=Q_i^TA_iQ_i$,只需

(1)计算Q_i的第1列(其平行于$A_i-\sigma_iI$的第一列，故只要通过规范化这个列向量就能得到).

(2)选择Q_i的其他列使Q_i正交而且A_{i+1}是不可约海森伯格阵.

于是由隐式Q定理知，已经正确地算出A_{i+1}，因为不计符号Q_i是唯一的(不计符号，因为改变Q_i的列的符号与将$A_i-\sigma_iI=Q_iR_i$改变为$(Q_iS_i)(S_iR_i)$是相同的，其中$S_i=\mathrm{diag}(\pm1,\cdots,\pm1)$. 于是$A_{i+1}=(S_iR_i)(Q_iS_i)+\sigma_iI=S_i(R_iQ_i+\sigma_iI)S_i$，这正好是改变$A_{i+1}$的行和列的符号的一个正交相似.)

隐式Q定理的证明. 假如$Q^TAQ=H$和$V^TAV=G$是不可约上海森伯格阵，Q和V是正交阵，Q和V的第1列相等. 设$(X)_i$表示X的第i列. 我们希望对所有$i>1$证明$(Q)_i=\pm(V)_i$或等价地$W\equiv V^TQ=\mathrm{diag}(\pm1,\cdots,\pm1)$.

因为$W=V^TQ$，所以得到$GW=GV^TQ=V^TAQ=V^TQH=WH$. 由$GW=WH$推出$G(W)_i=(GW)_i=(WH)_i=\sum_{j=1}^{i+1}h_{ji}(W)_j$,故$h_{i+1,i}(W)_{i+1}=G(W)_i-\sum_{j=1}^{i}h_{ji}(W)_j$. 因为$(W)_1=[1,0,\cdots,0]^T$而$G$是上海森伯格阵，所以我们可对$i$用归纳法证明$(W)_i$只有第1到第$i$个元素非零；即$W$是上三角阵. 因为$W$也是正交阵，所以$W$是对角阵$=\mathrm{diag}(\pm1,\cdots,\pm1)$. □

168

2. 隐式单个位移QR算法

为观察如何利用隐式Q定理从$A_0=A$计算A_1，利用一个5×5的例子.

例4.10

(1)选择

$$Q_1^T=\begin{bmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}\text{使}A_1\equiv Q_1^TAQ_1=\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ + & x & x & x & x \\ o & o & x & x & x \\ o & o & o & x & x \end{bmatrix}.$$

下面讨论如何选择c_1和s_1. 现在它们可能是任意的吉文斯旋转. 位置(3,1)中的+称为凸出部分(bulge)，为恢复到海森伯格型必须除掉.

(2) 选择

$$Q_2^{\mathrm{T}}=\begin{bmatrix}1&&&&\\&c_2&s_2&&\\&-s_2&c_2&&\\&&&1&\\&&&&1\end{bmatrix}\text{使 }Q_2^{\mathrm{T}}A_1=\begin{bmatrix}x&x&x&x&x\\x&x&x&x&x\\o&x&x&x&x\\o&o&x&x&x\\o&o&o&x&x\end{bmatrix}.\ \text{和 }A_2\equiv Q_2^{\mathrm{T}}A_1Q_2=\begin{bmatrix}x&x&x&x&x\\x&x&x&x&x\\o&x&x&x&x\\o&+&x&x&x\\o&o&o&x&x\end{bmatrix}.$$

凸出部分已经从(3,1)被“驱赶”到(4,2).

(3) 选择

$$Q_3^{\mathrm{T}}=\begin{bmatrix}1&&&&\\&1&&&\\&&c_3&s_3&\\&&-s_3&c_3&\\&&&&1\end{bmatrix}\text{使 }Q_3^{\mathrm{T}}A_2=\begin{bmatrix}x&x&x&x&x\\x&x&x&x&x\\o&x&x&x&x\\o&o&x&x&x\\o&o&o&x&x\end{bmatrix}\text{和 }A_3=Q_3^{\mathrm{T}}A_2Q_3=\begin{bmatrix}x&x&x&x&x\\x&x&x&x&x\\o&x&x&x&x\\o&o&x&x&x\\o&o&+&x&x\end{bmatrix}.$$

凸出部分已经从(4,2)被“驱赶”到(5,3).

169

(4) 选择

$$Q_4^{\mathrm{T}}=\begin{bmatrix}1&&&&\\&1&&&\\&&1&&\\&&&c_4&s_4\\&&&-s_4&c_4\end{bmatrix}\text{使 }Q_4^{\mathrm{T}}A_3=\begin{bmatrix}x&x&x&x&x\\x&x&x&x&x\\o&x&x&x&x\\o&o&x&x&x\\o&o&o&x&x\end{bmatrix}\text{和 }A_4=Q_4^{\mathrm{T}}A_3Q_4=\begin{bmatrix}x&x&x&x&x\\x&x&x&x&x\\o&x&x&x&x\\o&o&x&x&x\\o&o&o&x&x\end{bmatrix}.$$

这样我们已回到上海森伯格型.

合在一起 $Q^{\mathrm{T}}AQ$ 是上海森伯格型，其中

$$Q=Q_1Q_2Q_3Q_4=\begin{bmatrix}c_1&x&x&x&x\\s_1&x&x&x&x\\&s_2&x&x&x\\&&s_3&x&x\\&&&s_4&x\end{bmatrix},$$

故 Q 的第 1 列是 $[c_1,s_1,0,\cdots,0]^{\mathrm{T}}$，由隐式 Q 定理它唯一地决定 Q 的其他列(不计符号). 现在选择 Q 的第 1 列与 $A-\sigma I$ 的第 1 列 $[a_{11}-\sigma,a_{21},0,\cdots,0]^{\mathrm{T}}$ 成比例. 这意味着 Q 如要求的那样和 $A-\sigma I$ 的 QR 分解中的相同. ◇

$n\times n$ 阶矩阵一次隐式 QR 迭代的代价是 $6n^2+O(n)$.

3. 隐式双位移 QR 算法

本节描述如何同时用 σ 和 $\bar{\sigma}$ 位移保持实数运算. 对一个有效的实际执行过程来说这是基本的，但对算法的数学上理解不是基本的，第一次阅读时可以跳过.

依次用 σ 和 σ 位移的结果是

$$A_0-\sigma I=Q_1R_1,$$
$$A_1=R_1Q_1+\sigma I \qquad 故\ A_1=Q_1^{\mathrm T}A_0Q_1,$$
$$A_1-\sigma I=Q_2R_2,$$

170

$$A_2=R_2Q_2+\sigma I \qquad 故\ A_2=Q_2^{\mathrm T}A_1Q_2=Q_2^{\mathrm T}Q_1^{\mathrm T}A_0Q_1Q_2.$$

引理 4.5　我们可选择 Q_1 和 Q_2 使

(1) Q_1Q_2 是实的.

(2) 所以 A_2 是实的.

(3) Q_1Q_2 的第1列是容易计算的.

证明　因为 $Q_2R_2=A_1-\sigma I=R_1Q_1+(\sigma-\bar\sigma)I$，所以得到

$$\begin{aligned}Q_1Q_2R_2R_1&=Q_1(R_1Q_1+(\sigma-\bar\sigma)I)R_1\\&=Q_1R_1Q_1R_1+(\sigma-\bar\sigma)Q_1R_1\\&=(A_0-\sigma I)(A_0-\sigma I)+(\sigma-\bar\sigma)(A_0-\sigma I)\\&=A_0^2-2(\Re\sigma)A_0+|\sigma|^2I\equiv M.\end{aligned}$$

因此 $(Q_1Q_2)(R_2R_1)$ 是实阵 M 的 QR 分解，所以 Q_1Q_2 及 R_2R_1 可以被选择为实的. 这意味着 $A_2=(Q_1Q_2)^{\mathrm T}A(Q_1Q_2)$ 也是实的.

Q_1Q_2 的第1列与 $A_0^2-2(\Re\sigma)A_0+|\sigma^2|I$ 的第1列成比例，它是

$$\begin{bmatrix}a_{11}^2+a_{12}a_{21}-2(\Re\sigma)a_{11}+|\sigma|^2\\a_{21}(a_{11}+a_{22}-2(\Re\sigma))\\a_{21}a_{32}\\0\\\vdots\\0\end{bmatrix}.$$ □

利用隐式 Q 定理隐式地计算 Q_1Q_2 的其余列. 这个过程仍称为“凸出部分驱赶”，但现在凸出部分是 2×2，而不是 1×1.

例 4.11　下面是凸出部分驱赶的 6×6 阶矩阵例子.

(1) 选择 $Q_1^{\mathrm T}=\begin{bmatrix}\tilde Q_1^{\mathrm T}&0\\0&I\end{bmatrix}$，其中 $\tilde Q_1^{\mathrm T}$ 的第1列如上面给出的那样，故

$$Q_1^{\mathrm T}A=\begin{bmatrix}x&x&x&x&x&x\\x&x&x&x&x&x\\+&x&x&x&x&x\\o&o&x&x&x&x\\o&o&o&x&x&x\\o&o&o&o&x&x\end{bmatrix} 和\ A_1\equiv Q_1^{\mathrm T}AQ_1=\begin{bmatrix}x&x&x&x&x&x\\x&x&x&x&x&x\\+&x&x&x&x&x\\+&+&x&x&x&x\\o&o&o&x&x&x\\o&o&o&o&x&x\end{bmatrix}.$$

我们看出存在一个由加号表示的 2×2 凸出部分. [171]

(2) 选择一个豪斯霍尔德反射 $\boldsymbol{Q}_2^{\mathrm{T}}$，它只影响 $\boldsymbol{Q}_2^{\mathrm{T}}\boldsymbol{A}_1$ 的 2、3、4 行使 $\boldsymbol{A}_1$ 的(3,1)和(4,1)元素化为零(这意味着 $\boldsymbol{Q}_2^{\mathrm{T}}$ 除第 2 行到第 4 行，第 2 列到 4 列之外是单位阵)：

$$\boldsymbol{Q}_2^{\mathrm{T}}\boldsymbol{A}_1=\begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ o & x & x & x & x & x \\ o & + & x & x & x & x \\ o & o & o & x & x & x \\ o & o & o & o & x & x \end{bmatrix} \text{和 } \boldsymbol{A}_2\equiv\boldsymbol{Q}_2^{\mathrm{T}}\boldsymbol{A}_1\boldsymbol{Q}_2=\begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ o & x & x & x & x & x \\ o & + & x & x & x & x \\ o & + & + & x & x & x \\ o & o & o & o & x & x \end{bmatrix},$$

2×2 凸出部分被“驱赶”了一列.

(3) 选择一个豪斯霍尔德反射 $\boldsymbol{Q}_3^{\mathrm{T}}$，它只影响 $\boldsymbol{Q}_3^{\mathrm{T}}\boldsymbol{A}_2$ 的 3、4、5 行使 $\boldsymbol{A}_2$ 的(4,2)和(5,2)元素化为零(意味着 $\boldsymbol{Q}_3^{\mathrm{T}}$ 除第 3 行到第 5 行，第 3 列到第 5 列之外是单位阵)：

$$\boldsymbol{Q}_3^{\mathrm{T}}\boldsymbol{A}_2=\begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ o & x & x & x & x & x \\ o & o & x & x & x & x \\ o & o & + & x & x & x \\ o & o & o & o & x & x \end{bmatrix} \text{和 } \boldsymbol{A}_3\equiv\boldsymbol{Q}_3^{\mathrm{T}}\boldsymbol{A}_2\boldsymbol{Q}_3=\begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ o & x & x & x & x & x \\ o & o & x & x & x & x \\ o & o & + & x & x & x \\ o & o & + & + & x & x \end{bmatrix}.$$

(4) 选择一个豪斯霍尔德反射 $\boldsymbol{Q}_4^{\mathrm{T}}$，它只影响 $\boldsymbol{Q}_4^{\mathrm{T}}\boldsymbol{A}_3$ 的 4、5、6 行，使 $\boldsymbol{A}_3$ 的(5,3)和(6,3)元素化为零(这意味着 $\boldsymbol{Q}_4^{\mathrm{T}}$ 除第 4 行到第 6 行，第 4 列到第 6 列之外是单位阵)：

$$\boldsymbol{A}_4\equiv\boldsymbol{Q}_4^{\mathrm{T}}\boldsymbol{A}_3\boldsymbol{Q}_4=\begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ o & x & x & x & x & x \\ o & o & x & x & x & x \\ o & o & o & x & x & x \\ o & o & o & + & x & x \end{bmatrix}.$$

(5) 选择

$$\boldsymbol{Q}_5^{\mathrm{T}}=\left[\begin{array}{cccc|cc} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ \hline & & & & c & s \\ & & & & -s & c \end{array}\right] \text{使 } \boldsymbol{A}_5=\boldsymbol{Q}_5^{\mathrm{T}}\boldsymbol{A}_4\boldsymbol{Q}_5=\begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ o & x & x & x & x & x \\ o & o & x & x & x & x \\ o & o & o & x & x & x \\ o & o & o & o & x & x \end{bmatrix}. \qquad \diamond$$

[172]

4. 选择 QR 算法的位移

为完全确定一个具有单位移或者是双位移的海森伯格 QR 迭代，我们必须选择位

移 σ(和 $\bar{\sigma}$). 从4.4.4节末尾记得导致渐近二次收敛于一个实特征值的单位移的适当选择是 A 的右下方的元素 $\sigma = a_{n,n}$. 推广到双位移是利用 Francis 位移(Francis shift), 这意味着 σ 和 $\bar{\sigma}$ 是 A_i 的右下角 2×2 阶矩阵: $\begin{bmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-1} & a_{n,n} \end{bmatrix}$的特征值. 这将使我们收敛于右下角2×2阶矩阵的两个实特征值或者具有复共轭特征值的单个2×2阶矩阵. 当接近于收敛时, 预期 $a_{n-1,n-2}$(以及可能是 $a_{n,n-1}$)是小的使得2×2阶矩阵的特征值是 A 的特征值的好的近似值. 确实, 可以证明这个选择导致二次渐近收敛. 这意味着一旦 $a_{n-1,n-2}$(以及可能是 $a_{n,n-1}$)足够小, 它的值每步将平方且很快逼近于零. 实际上, 这项工作如此之好, 几乎对所有的矩阵每个特征值收敛平均只需二次 QR 迭代. 这就证明称 QR 迭代为“直接”法是恰当的.

实际上, 用 Francis 位移的 QR 迭代收敛可能失败(确实, 它保持$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$不变). 故使用了数十年的实际算法每十次位移若不出现收敛, 就会有一个“特殊的位移”. 最近还发现此算法不收敛的少量矩阵[25,65]; 在矩阵

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & h & 0 \\ 0 & -h & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

的一个小邻域中的矩阵, 构成这样一组矩阵, 其中 h 是几千倍的机器精度. 故另一个“特殊的位移”最近被加入到算法中以修补这种情况. 但是找一个对一切矩阵都保证快速收敛的位移策略仍然是一个未决的问题.

4.5　其他的非对称特征值问题

4.5.1　正则矩阵束和魏尔斯特拉斯典范型

标准特征值问题询问对哪个标量 z 矩阵 $A - zI$ 奇异; 这个标量是特征值. 这个概念推广至几个重要的情形.

173

定义4.6　$A - \lambda B$ 称为一个矩阵束, 或简称束, 其中 A 和 B 是 $m \times n$ 阶矩阵, 而 λ 是一个不确定的值, 不是一个特别的数值.

定义4.7　若 A 和 B 是方的且 $\det(A - \lambda B)$ 不恒等于零, 则束 $A - \lambda B$ 称为正则的. 否则它称为奇异的. 当 $A - \lambda B$ 正则时, $p(\lambda) \equiv \det(A - \lambda B)$ 称为 $A - \lambda B$ 的特征多项式, $A - \lambda B$ 的特征值被定义为

(1) $p(\lambda) = 0$ 的根;

(2) 当 $\det(p)<n$ 时为∞(具有重数 $n-\deg(p)$).

例 4.12 设

$$A-\lambda B=\begin{bmatrix}1 & & \\ & 1 & \\ & & 0\end{bmatrix}-\lambda\begin{bmatrix}2 & & \\ & 0 & \\ & & 1\end{bmatrix}.$$

则 $p(\lambda)=\det(A-\lambda B)=(1-2\lambda)\cdot(1-0\lambda)\cdot(0-\lambda)=(2\lambda-1)\lambda$,故特征值为 $\lambda=1/2,0$ 和∞. ◇

矩阵束自然地产生于物理系统的许多数学模型中，我们给出下列一些例子. 下面的命题把正则矩阵束 $A-\lambda B$ 的特征值和单个矩阵的特征值联系起来.

命题 4.6 设 $A-\lambda B$ 是正则的. 若 B 非奇异，则 $A-\lambda B$ 的所有特征值是有限的，并且与 AB^{-1} 或 $B^{-1}A$ 的特征值相同. 若 B 奇异，则 $A-\lambda B$ 有重数为 $n-\text{rank}(B)$ 的特征值∞. 若 A 非奇异，则 $A-\lambda B$ 的特征值与 $A^{-1}B$ 或 BA^{-1} 的特征值的倒数相同，其中 $A^{-1}B$ 的一个零特征值对应于 $A-\lambda B$ 的一个无穷特征值.

证明 若 B 非奇异且 λ' 是一个特征值，则 $0=\det(A-\lambda' B)=\det(AB^{-1}-\lambda' I)=\det(B^{-1}A-\lambda' I)$，故 λ' 也是 AB^{-1} 和 $B^{-1}A$ 的一个特征值. 若 B 奇异，则取 $p(\lambda)=\det(A-\lambda B)$,记 B 的 SVD 为 $B=U\Sigma V^T$,代入 $p(\lambda)$ 得到

$$p(\lambda)=\det(A-\lambda U\Sigma V^T)=\det(U(U^TAV-\lambda\Sigma)V^T)=\pm\det(U^TAV-\lambda\Sigma).$$

因为 $\text{rank}(B)=\text{rank}(\Sigma)$，所以在 $U^TAV-\lambda\Sigma$ 中只出现 $\text{rank}(B)$ 个 λ，故多项式 $\det(U^TAV-\lambda\Sigma)$ 的次数为 $\text{rank}(B)$.

若 A 非奇异，则 $\det(A-\lambda B)=0$ 当且仅当 $\det(I-\lambda A^{-1}B)=0$ 或 $\det(I-\lambda BA^{-1})=0$. 这个等式仅当 $\lambda\neq 0$ 且 $1/\lambda$ 是 $A^{-1}B$ 或 BA^{-1} 的一个特征值时成立. □ 174

定义 4.8 设 λ' 是正则束 $A-\lambda B$ 的一个有限的特征值. 则当 $(A-\lambda' B)x=0$，或等价地 $Ax=\lambda' Bx$ 时，$x\neq 0$ 是一个右特征向量. 若 $\lambda'=\infty$ 是一个特征值且 $Bx=0$，则 x 是一个右特征向量. $A-\lambda B$ 的一个左特征向量是 $(A-\lambda B)^*$ 的一个右特征向量.

例 4.13 考虑例 4.12 中的束 $A-\lambda B$. 因为 A 和 B 是对角阵，所以右和左特征向量正好是单位阵的列. ◇

例 4.14 考虑例 4.1 的阻尼质点－弹簧系统. 从这个问题自然地产生两个矩阵束. 首先可以记特征值问题

$$Ax=\begin{bmatrix}-M^{-1}B & -M^{-1}K \\ I & 0\end{bmatrix}x=\lambda x$$

为

$$\begin{bmatrix}-B & -K \\ I & 0\end{bmatrix}x=\lambda\begin{bmatrix}M & 0 \\ 0 & I\end{bmatrix}x.$$

若 M 非常病态，以致很难准确地计算 $M^{-1}B$ 和 $M^{-1}K$，这可能是一个较好的数学表述.

其次，通常考虑 $B=0$（无阻尼）情况，故原来的微分方程是 $M\ddot{x}(t)+Kx(t)=0$. 寻找形如 $x_i(t)=e^{\lambda_i t}x_i(0)$ 的解. 我们得到 $\lambda_i^2 e^{\lambda_i t}Mx_i(0)+e^{\lambda_i t}Kx_i(0)=0$ 或 $\lambda_i^2 Mx_i(0)+Kx_i(0)=0$. 换言之，$-\lambda_i^2$ 是束 $K-\lambda M$ 的一个特征值而 $x_i(0)$ 是束 $K-\lambda M$ 的一个右特征向量. 因为已假定 M 非奇异，所以也存在 $M^{-1}K$ 的特征值和特征向量. ◇

在实践中也自然地产生无限特征值. 例如，在本节的后面将指出无限特征值如何对应于线性约束常微分方程或微分－代数方程（differential-algebraic equations）描述的系统中的脉冲响应（impulse response）[41]. 关于矩阵束应用于计算几何和计算机绘图可见问题4.16.

记得单个矩阵 A 的特征值问题的全部理论和算法依赖于求 A 的一个相似变换 $S^{-1}AS$，这是一个比 A “更简单的” 形式. 下面的定义指出如何把相似性的概念推广到矩阵束中，然后指出如何把若尔当型和舒尔型推广到束中去.

定义4.9　设 P_L 和 P_R 是非奇异阵. 则束 $A-\lambda B$ 和 $P_LAP_R-\lambda P_LBP_R$ 称为等价的.

175

命题4.7　等价的正则束 $A-\lambda B$ 和 $P_LAP_R-\lambda P_LBP_R$ 有相同的特征值. 向量 x 是 $A-\lambda B$ 的一个右特征向量当且仅当 $P_R^{-1}x$ 是 $P_LAP_R-\lambda P_LBP_R$ 的一个右特征向量. 向量 y 是 $A-\lambda B$ 的一个左特征向量当且仅当 $(P_L^*)^{-1}y$ 是 $P_LAP_R-\lambda P_LBP_R$ 的一个左特征向量.

证明

$$\det(A-\lambda B)=0 \quad \text{当且仅当} \quad \det(P_L(A-\lambda B)P_R)=0.$$
$$(A-\lambda B)x=0 \quad \text{当且仅当} \quad P_L(A-\lambda B)P_RP_R^{-1}x=0.$$
$$(A-\lambda B)^*y=0 \quad \text{当且仅当} \quad P_R^*(A-\lambda B)^*P_L^*(P_L^*)^{-1}y=0.$$ □

下列定理将若尔当典范型推广至正则矩阵束.

定理4.10　魏尔斯特拉斯典范型.　设 $A-\lambda B$ 正则. 则存在非奇异的 P_L 和 P_R 使得

$$P_L(A-\lambda B)P_R=\operatorname{diag}(J_{n_1}(\lambda_1)-\lambda I_{n_1},\cdots,J_{n_k}(\lambda_{n_k})-\lambda I_{n_k},N_{m_1},\cdots,N_{m_r}),$$

其中 $J_{n_i}(\lambda_i)$ 是具有特征值 λ_i 的 $n_i\times n_i$ 若尔当块.

$$J_{n_i}(\lambda_i)=\begin{bmatrix}\lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i\end{bmatrix},$$

而 N_{m_i} 是 “具有重数 m_i 的 $\lambda=\infty$ 的若尔当块”

$$N_{m_i}=\begin{bmatrix}1 & \lambda & & \\ & 1 & \ddots & \\ & & \ddots & \lambda \\ & & & 1\end{bmatrix}=I_{m_i}-\lambda J_{m_i}(0).$$

证明见[110].

1. 应用若尔当型和魏尔斯特拉斯型于微分方程

考虑线性微分方程 $\dot{x}(t)=Ax(t)+f(t),x(0)=x_0$. 由 $x(t)=e^{At}x_0+\int_0^t e^{A(t-\tau)}f(\tau)\mathrm{d}\tau$

给出一个显式解. 若我们知道若尔当型 $A = SJS^{-1}$，则可以将微分方程中的变量改变为 $y(t) = S^{-1}x(t)$ 得到 $\dot{y}(t) = Jy(t) + S^{-1}f(t)$，具有解 $y(t) = e^{Jt}y_0 + \int_0^t e^{J(t-\tau)}S^{-1}f(\tau)\,d\tau$. 有一个计算 e^{Jt} 或具有若尔当型 J 的矩阵的其他函数 $f(J)$ 的显式公式(数值上不应该利用这个公式！作为一个较好的算法的基础，见问题 4.4)假如 f 由其泰勒级数 $f(z) = \sum_{i=0}^{\infty} \frac{f^{(i)}(0)z^i}{i!}$ 给出，而 J 是单个若尔当块 $J = \lambda I + N$，其中矩阵 N 在第 1 条上对角线上为 1，其余地方为 0. 于是

176

$$
\begin{aligned}
f(J) &= \sum_{i=0}^{\infty} \frac{f^{(i)}(0)(\lambda I + N)^i}{i!} \\
&= \sum_{i=0}^{\infty} \frac{f^{(i)}(0)}{i!} \sum_{j=0}^{i} \binom{i}{j} \lambda^{i-j} N^j \qquad \text{由二项式定理} \\
&= \sum_{j=0}^{\infty} \sum_{i=j}^{\infty} \frac{f^{(i)}(0)}{i!} \binom{i}{j} \lambda^{i-j} N^j \qquad \text{颠倒求和的次序} \\
&= \sum_{j=0}^{n-1} N^j \sum_{i=j}^{\infty} \frac{f^{(i)}(0)}{i!} \binom{i}{j} \lambda^{i-j},
\end{aligned}
$$

其中在最后的等式中我们利用了 $N^j = 0$ 对 $j > n-1$. 注意 N^j 在第 j 条上对角线上为 1 其余地方为 0. 最后，注意 $\sum_{i=j}^{\infty} \frac{f^{(i)}(0)}{i!}\binom{i}{j}\lambda^{i-j}$ 是 $f^{(j)}(\lambda)/j!$ 的泰勒展开. 因此

$$
f(J) = f\left(\begin{bmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix}^{n \times n} \right) = \sum_{j=0}^{n-1} \frac{N^j f^{(j)}(\lambda)}{j!}
$$

$$
= \begin{bmatrix} f(\lambda) & f'(\lambda) & \frac{f''(\lambda)}{2!} & \cdots & \frac{f^{(n-1)}(\lambda)}{(n-1)!} \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \frac{f''(\lambda)}{2!} \\ & & & \ddots & f'(\lambda) \\ & & & & f(\lambda) \end{bmatrix} \tag{4.6}
$$

故 $f(J)$ 是在第 j 条上对角线上为 $f^{(j)}(\lambda)/j!$ 的上三角阵.

为求解更一般的问题 $B\dot{x} = Ax + f(t)$，$A - \lambda B$ 正则，我们利用魏尔斯特拉斯典范型：设 $P_L(A - \lambda B)P_R$ 是魏尔斯特拉斯型，改写方程为 $P_L B P_R P_R^{-1}\dot{x} = P_L A P_R P_R^{-1}x + P_L f(t)$. 设 $P_R^{-1}x = y$ 及 $P_L f(t) = g(t)$. 现在问题已被分解成子问题：

$$\begin{bmatrix} I_{n_1} & & & & & \\ & \ddots & & & & \\ & & I_{n_k} & & & \\ & & & J_{m_1}(0) & & \\ & & & & \ddots & \\ & & & & & J_{m_r}(0) \end{bmatrix} \dot{y} = \begin{bmatrix} J_{n_1}(\lambda_1) & & & & & \\ & \ddots & & & & \\ & & J_{n_k}(\lambda_k) & & & \\ & & & I_{m_1} & & \\ & & & & \ddots & \\ & & & & & I_{m_r} \end{bmatrix} y + g.$$

177

每个子问题 $\dot{\tilde{y}} = J_{n_i}(\lambda_i)\tilde{y} + \tilde{g}(t) \equiv J\tilde{y} + \tilde{y}(t)$ 是一个如上面一样的标准线性 ODE,具有解

$$\tilde{y}(t) = \tilde{y}(0)e^{Jt} + \int_0^t e^{J(t-\tau)}\tilde{g}(\tau)d\tau.$$

从最后的方程开始通过向后回代得到 $J_m(0)\dot{\tilde{y}} = \tilde{y} + \tilde{g}(t)$ 的解:记 $J_m(0)\dot{\tilde{y}} = \tilde{y} + \tilde{g}(t)$ 为

$$\begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix} \begin{bmatrix} \dot{\tilde{y}}_1 \\ \vdots \\ \vdots \\ \dot{\tilde{y}}_m \end{bmatrix} = \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \vdots \\ \tilde{y}_m \end{bmatrix} + \begin{bmatrix} \tilde{g}_1 \\ \vdots \\ \vdots \\ \tilde{g}_m \end{bmatrix}.$$

第 m 个(最后的)方程为 $0 = \tilde{y}_m + \tilde{g}_m$ 或 $\tilde{y}_m = -\tilde{g}_m$. 第 i 个方程为 $\dot{\tilde{y}}_{i+1} = \tilde{y}_i + \tilde{g}_i$,故 $\tilde{y}_i = \dot{\tilde{y}}_{i+1} - \tilde{g}_i$, 因而

$$\tilde{y}_i = \sum_{k=i}^{m} -\frac{d^{k-i}}{dt^{k-i}}\tilde{g}_k(t).$$

所以解依赖于 $\tilde{g}$ 的导数而不是在通常的 ODE 中依赖于 g 的一个积分. 从而一个连续而不可微的 $\tilde{g}$ 可能引起解中的不连续性; 这有时称为一个脉冲响应并且仅当有无限特征值时出现. 此外有一个连续解 $\tilde{y}$ 必须在 $t=0$ 处满足某些相容性条件:

$$y_i(0) = \sum_{k=m}^{i} -\frac{d^{k-i}}{dt^{k-i}}g_k(0).$$

基于时间 - 步长的求解这种微分代数方程或具有代数约束的 ODE 数值方法在 [41] 中描述.

2. 正则束的广义舒尔型

正如我们不能稳定地计算若尔当型一样, 我们不能稳定地计算它的推广魏尔斯特拉斯型. 因而, 我们计算广义的舒尔型.

178

定理 4.11 广义舒尔型. 设 $A - \lambda B$ 正则. 则存在酉阵 Q_L 和 Q_R 使得 $Q_L A Q_R = T_A$ 和 $Q_L B Q_R = T_B$ 都是上三角的. 于是 $A - \lambda B$ 的特征值是 T_A 和 T_B 的对角元之比 $T_{A_{ii}}/T_{B_{ii}}$.

证明 证明非常类似于通常的舒尔型. 设 λ' 是一个特征值而 x 是一个单位右特征向量:

$\|x\|_2=1$. 因为 $Ax-\lambda'Bx=0$,Ax 和 Bx 两者都是同一个单位向量 y 的倍数(即使 Ax 和 Bx 之一为零). 今设 $X=[x, \widetilde{X}]$ 和 $Y=[y, \widetilde{Y}]$ 是分别具有第1列 x 和 y 的酉阵. 则由构造

$$Y^*AX=\begin{bmatrix}\widetilde{a}_{11} & \widetilde{a}_{12}\\ 0 & \widetilde{A}_{22}\end{bmatrix}\text{和 } Y^*BX=\begin{bmatrix}\widetilde{b}_{11} & \widetilde{b}_{12}\\ 0 & \widetilde{B}_{22}\end{bmatrix}.$$

对 $\widetilde{A}_{22}-\lambda\widetilde{B}_{22}$ 归纳地应用这个过程. □

若 A 和 B 为实的,则也存在一个广义的实舒尔型:实正交的 Q_L 和 Q_R,其中 Q_LAQ_R 是准上三角的而 Q_LBQ_R 是上三角的.

QR 算法及其所有的改进推广到计算广义(实)舒尔型;它称为 QZ 算法,并可利用 LAPACK 中的子程序 sgges. 在 Matlab 中利用指令 eig(A,B)来实现.

3. 定束

在实践中经常出现的一个较简单的情况是束 $A-\lambda B$,其中 $A=A^T$,$B=B^T$ 且 B 是正定的. 这样的束称为定束(definite pencil).

定理4.12 设 $A=A^T$ 且设 $B=B^T$ 是正定的. 则存在一个非奇异实阵 X 使得 $X^TAX=\text{diag}(\alpha_1,\cdots,\alpha_n)$ 和 $X^TBX=\text{diag}(\beta_1,\cdots,\beta_n)$. 特别地,所有的特征值 α_i/β_i 是实的和有限的.

证明 给出的证明实际上是用于求解问题的算法:

(1) 设 $LL^T=B$ 是楚列斯基分解.

(2) 设 $H=L^{-1}AL^{-T}$;注意 H 是对称的.

(3) 设 $H=Q\Lambda Q^T$,Q 为正交的,Λ 为实对角的.

则 $X=L^{-T}Q$ 满足 $X^TAX=Q^TL^{-1}AL^{-T}Q=\Lambda$ 和 $X^TBX=Q^TL^{-1}BL^{-T}Q=I$. □

注意当对某些标量 α 和 β,$\alpha A+\beta B$ 正定时,定理也是成立的.

这个问题的软件可以利用 LAPACK 程序 ssygv.

例4.15 考虑来自例4.14的束 $K-\lambda M$. 这是一个定束,因为刚度矩阵 K 对称而质量矩阵 M 对称正定. 事实上在这个非常简单的例子中 K 是三对角的而 M 是对角的,故 M 的楚列斯基因子 L 也是对角的,而 $H=L^{-1}KL^{-T}$ 也是对称三对角的. 在第5章中我们将考虑对称三对角特征问题的各种算法. ◇ [179]

4.5.2 奇异矩阵束和克罗内克典范型

现在考虑奇异束 $A-\lambda B$. 记得当 A 和 B 不是方阵或者它们是方阵但对所有 λ 的值 $\det(A-\lambda B)=0$ 时 $A-\lambda B$ 奇异. 下例指出把特征值定义推广到这种情况时需要谨慎.

例4.16 设 $A=\begin{bmatrix}1&0\\0&0\end{bmatrix}$,$B=\begin{bmatrix}1&0\\0&0\end{bmatrix}$. 则作任意小的改变得到 $A'=\begin{bmatrix}1&\epsilon_1\\ \epsilon_2&0\end{bmatrix}$ 和 $B'=\begin{bmatrix}1&\epsilon_3\\ \epsilon_4&0\end{bmatrix}$,特征值变为 ϵ_1/ϵ_3 和 ϵ_2/ϵ_4,这可能是任意的复数. 故特征值无限地敏感. ◇

尽管这种极端的敏感性,奇异束还是被用于某些物理系统建模中,正如我们下

面描述的那样.

继续指出如何推广若尔当型和魏尔斯特拉斯型于奇异束. 除若尔当块和"无限若尔当"块外, 我们在典范型中得到两个新的"奇异块".

定理 4.13 克罗内克典范型. 设 A 和 B 是任意的 $m\times n$ 阶矩阵. 则存在非奇异方阵 P_L 和 P_R 使得 $P_LAP_R-\lambda P_LBP_R$ 是具有四类块的块对角阵:

$$J_m(\lambda')-\lambda I=\begin{bmatrix}\lambda'-\lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda'-\lambda\end{bmatrix},\quad m\times m \text{ 若尔当块}$$

$$N_m=\begin{bmatrix}1 & \lambda & & \\ & \ddots & \ddots & \\ & & \ddots & \lambda \\ & & & 1\end{bmatrix},\quad m\times m \text{ 若尔当块}, \lambda=\infty$$

180

$$L_m=\begin{bmatrix}1 & \lambda & & \\ & \ddots & \ddots & \\ & & 1 & \lambda\end{bmatrix},\quad m\times(m+1) \text{ 右奇异块}$$

$$L_m^T=\begin{bmatrix}1 & & \\ \lambda & \ddots & \\ & \ddots & 1 \\ & & \lambda\end{bmatrix},\quad (m+1)\times m \text{ 左奇异块}$$

称 L_m 为右奇异块是因为它对一切 λ 有一个右零向量 $[\lambda^m,-\lambda^{m-1},\cdots,\pm 1]$. L_m^{T} 有一个类似的左零向量.

证明见[110].

正如在上节中舒尔型推广到正则矩阵束一样, 它同样能够推广到任意的奇异束. 关于典范型、扰动理论和软件见[27,79,246].

奇异束被用于系统和控制中产生的模型系统. 我们给出两个例子.

1. 应用克罗内克型于微分方程

假如要求解 $B\dot{x}=Ax+f(t)$, 其中 $A-\lambda B$ 是奇异束. 记 $P_LBP_RP_R^{-1}\dot{x}=P_LAP_RP_R^{-1}x+P_Lf(t)$ 分解问题成独立的块. 有四种类型, 每块都是克罗内克型中的一种类型. 当考虑正则束和魏尔斯特拉斯型时已处理了 $J_m(\lambda')-\lambda I$ 和 N_m 块,故我们只需考虑 L_m 块和 L_m^{T} 块. 从 L_m 块得到

$$\begin{bmatrix}0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{y}_1\\ \vdots \\ \dot{y}_{m+1}\end{bmatrix}=\begin{bmatrix}1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0\end{bmatrix}\begin{bmatrix}y_1\\ \vdots \\ y_{m+1}\end{bmatrix}+\begin{bmatrix}g_1\\ \vdots \\ g_m\end{bmatrix}$$

或

$$\begin{aligned}
\dot{y}_2 &= y_1 + g_1 \quad \text{或} \quad y_2(t) = y_2(0) + \int_0^t (y_1(\tau) + g_1(\tau))\mathrm{d}\tau, \\
\dot{y}_3 &= y_2 + g_2 \quad \text{或} \quad y_3(t) = y_3(0) + \int_0^t (y_2(\tau) + g_2(\tau))\mathrm{d}\tau, \\
&\vdots \\
\dot{y}_{m+1} &= y_m + g_m \quad \text{或} \quad y_{m+1}(t) = y_{m+1}(0) + \int_0^t (y_m(\tau) + g_m(\tau))\mathrm{d}\tau,
\end{aligned}$$

这意味着可选择 y_1 为一个任意的可积函数并利用上面的递推关系得到一个解. 这是因为未知量比方程多一个, 所以此 ODE 是亚定的(underdetermined). 从 L_m^{T} 块得到 181

$$\begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & 0 \\ & & 1 \end{bmatrix} \begin{bmatrix} \dot{y}_1 \\ \vdots \\ \dot{y}_m \end{bmatrix} = \begin{bmatrix} 1 & & \\ 0 & \ddots & \\ & \ddots & 1 \\ & & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} + \begin{bmatrix} g_1 \\ \vdots \\ g_{m+1} \end{bmatrix}$$

或

$$\begin{aligned}
0 &= y_1 + g_1, \\
\dot{y}_1 &= y_2 + g_2, \\
&\vdots \\
\dot{y}_{m-1} &= y_m + g_m, \\
\dot{y}_m &= g_{m+1}.
\end{aligned}$$

从第一个方程开始求解得到

$$\begin{aligned}
y_1 &= -g_1, \\
y_2 &= -g_2 - \dot{g}_1, \\
&\vdots \\
y_m &= -g_m - \dot{g}_{m-1} - \cdots - \frac{\mathrm{d}^{m-1}}{\mathrm{d}t^{m-1}} g_1
\end{aligned}$$

以及相容性条件(consistency condition) $g_{m+1} = -\dot{g}_m - \cdots - \frac{\mathrm{d}^m}{\mathrm{d}t^m} g_1$. 因此除非 g_i 满足这个方程, 否则无解. 这里方程比未知量多一个, 而且子问题是超定(overdetermined)的.

2. 应用克罗内克型于系统论和控制论

$\dot{x}(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$ 的可控子空间(controllable subspace)是一个空间, 其中系统状态(system state) $x(t)$ 可通过选择在 $x(0) = 0$ 起始的控制输入(control input) $u(t)$ 来"控制". 这个方程用于建立(反馈)控制系统模型, 其中 $u(t)$ 由控制系统工程师选择使 $x(t)$ 有某些期望的性质, 例如有界性. 从

$$x(t) = \int_0^t \mathrm{e}^{\boldsymbol{A}(t-\tau)} \boldsymbol{B}u(\tau)\mathrm{d}\tau = \int_0^t \sum_{i=0}^{\infty} \frac{(t-\tau)^i}{i!} \boldsymbol{A}^i \boldsymbol{B}u(\tau)\mathrm{d}\tau = \sum_{i=0}^{\infty} \boldsymbol{A}^i \boldsymbol{B} \int_0^t \frac{(t-\tau)^i}{i!} u(\tau)\mathrm{d}\tau$$

可以证明可控空间是 $\mathrm{span}\{[\boldsymbol{B}, \boldsymbol{AB}, \boldsymbol{A}^2\boldsymbol{B}, \cdots, \boldsymbol{A}^{n-1}\boldsymbol{B}]\}$; 在这个空间外边的 $x(t)$ 的任何

分量不能通过改变 $u(t)$ 来控制. 为了在实践中计算这个空间, 人们把 QR - 类算法应用于奇异束 $[\boldsymbol{B}, \boldsymbol{A}-\lambda \boldsymbol{I}]$, 以便确定建立的物理系统模型事实上是否由输入 $u(t)$ 控制. 详细情况见[78,246,247].

182

4.5.3 非线性特征值问题

最后, 考虑非线性特征值问题或矩阵多项式

$$\sum_{i=0}^{d} \lambda^i \boldsymbol{A}_i = \lambda^d \boldsymbol{A}_d + \lambda^{d-1} \boldsymbol{A}_{d-1} + \cdots + \lambda \boldsymbol{A}_1 + \boldsymbol{A}_0. \tag{4.7}$$

为简单起见假定 $\boldsymbol{A}_i$ 是 $n \times n$ 阵而 $\boldsymbol{A}_d$ 非奇异.

定义 4.10　矩阵多项式(4.7)的特征多项式是 $p(\lambda)=\det(\sum_{i=0}^{d} \lambda^i \boldsymbol{A}_i)$. $p(\lambda)=0$ 的根定义为特征值. 可以确定 $p(\lambda)$ 为 $d \cdot n$ 次, 故有 $d \cdot n$ 个特征值. 假如 γ 是一个特征值. 一个满足 $\sum_{i=0}^{d} \gamma^i \boldsymbol{A}_i x=0$ 的非零向量 $\boldsymbol{x}$ 是关于 γ 的右特征向量. 左特征向量 $\boldsymbol{y}$ 类似地由 $\sum_{i=0}^{d} \gamma^i \boldsymbol{y}^* \boldsymbol{A}_i=0$ 确定.

例 4.17　再次考虑例 4.1. 在(4,3)式中出现的 ODE 是 $\boldsymbol{M}\ddot{x}(t)+\boldsymbol{B}\dot{x}(t)+\boldsymbol{K}x(t)=0$. 若寻找形如 $x(t)=e^{\lambda_i t}x_i(0)$ 的解, 我们得到 $e^{\lambda_i t}(\lambda_i^2 \boldsymbol{M}x_i(0)+\lambda_i \boldsymbol{B}x_i(0)+\boldsymbol{K}x_i(0))=0$, 或 $\lambda_i^2 \boldsymbol{M}x_i(0)+\lambda_i \boldsymbol{B}x_i(0)+\boldsymbol{K}x_i(0)=0$. 因此 λ_i 是一个特征值而 $x_i(0)$ 是矩阵多项式 $\lambda^2 \boldsymbol{M}+\lambda \boldsymbol{B}+\boldsymbol{K}$ 的一个特征向量.　◇

因为我们假定 $\boldsymbol{A}_d$ 非奇异, 所以可用 $\boldsymbol{A}_d^{-1}$ 相乘得到等价的问题 $\lambda^d \boldsymbol{I}+\boldsymbol{A}_d^{-1}\boldsymbol{A}_{d-1}\lambda^{d-1}+\cdots+\boldsymbol{A}_d^{-1}\boldsymbol{A}_0$. 所以, 为保持记号简单, 将假定 $\boldsymbol{A}_d=\boldsymbol{I}$(一般情况见 4.6 节). 在最简单的情况, 每个 A_i 是 1×1 阶矩阵, 即一个标量, 原来的矩阵多项式等于特征多项式.

可以利用把一个高阶 ODE 变为一阶 ODE 的一个类似的诀窍, 把求矩阵多项式的特征值问题转换为求一个标准的特征值问题. 首先考虑最简单情况 $n=1$, 其中每个 A_i 是标量. 假如 γ 是一个根. 则向量 $\boldsymbol{x}'=[\gamma^{d-1}, \gamma^{d-2}, \cdots, \gamma, 1]^{\mathrm{T}}$ 满足

$$\boldsymbol{C}\boldsymbol{x}' \equiv \begin{bmatrix} -A_{d-1} & -A_{d-2} & \cdots & \cdots & \cdots & -A_0 \\ 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{bmatrix} x' = \begin{bmatrix} -\sum_{i=0}^{d-1} \gamma^i A_i \\ \gamma^{d-1} \\ \vdots \\ \gamma^2 \\ \gamma \end{bmatrix} = \begin{bmatrix} \gamma^d \\ \gamma^{d-1} \\ \vdots \\ \gamma^2 \\ \gamma \end{bmatrix} = \gamma \boldsymbol{x}'.$$

183

因此 $\boldsymbol{x}'$ 是一个特征向量而 γ 是矩阵 $\boldsymbol{C}$ 的一个特征值, 矩阵 $\boldsymbol{C}$ 称为多项式(4.7)的友阵.

(求多项式之根的 Matlab 程序 roots 对友阵 $\boldsymbol{C}$ 应用 4.4.8 节的海森伯格 QR 迭代, 即使代价昂贵, 它仍然是目前最可靠的已知方法之一[100,117,241]. 比较便宜的另一些方法在下面讨论.)

当 $\boldsymbol{A}_i$ 为矩阵时, 实施同样的想法. $\boldsymbol{C}$ 变成一个 $(n\cdot d)\times(n\cdot d)$ 块友阵, 其中第

一行下面的 1 和 0 分别变成 $n\times n$ 的单位阵和零阵. $\boldsymbol{x}'$ 也变成

$$\boldsymbol{x}'=\begin{bmatrix}\gamma^{d-1}x\\ \gamma^{d-2}x\\ \vdots\\ \gamma x\\ x\end{bmatrix},$$

其中 $\boldsymbol{x}$ 是矩阵多项式的一个右特征向量. 再次出现 $\boldsymbol{C}\boldsymbol{x}'=\gamma\boldsymbol{x}'$.

例 4.18　再次回到 $\lambda^2\boldsymbol{M}+\lambda\boldsymbol{B}+\boldsymbol{K}$, 首先把它转换成 $\lambda^2+\lambda\boldsymbol{M}^{-1}\boldsymbol{B}+\boldsymbol{M}^{-1}\boldsymbol{K}$ 然后转换成友阵

$$\boldsymbol{C}=\begin{bmatrix}-\boldsymbol{M}^{-1}\boldsymbol{B} & -\boldsymbol{M}^{-1}\boldsymbol{K}\\ \boldsymbol{I} & 0\end{bmatrix}.$$

这与例 4.1 的(4.4)式中矩阵 $\boldsymbol{A}$ 相同. ◇

最后, 问题 4.16 指出如何利用矩阵多项式求解计算几何中的一个问题.

4.6 小　结

下面的一览表概述本章中所描述的所有典范型、算法、它们的代价及对 ODE 的应用. 它也包括算法利用对称性的建议, 虽然这些将在下一章中更详尽地加以讨论. 稀疏矩阵的算法在第 7 章中讨论.

- $\boldsymbol{A}-\lambda\boldsymbol{I}$

—若尔当型: 对某个非奇异的 $\boldsymbol{S}$,

$$\boldsymbol{A}-\lambda\boldsymbol{I}=\boldsymbol{S}\cdot\mathrm{diag}\left(\cdots,\begin{bmatrix}\lambda_i-\lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1\\ & & & \lambda_i-\lambda\end{bmatrix}^{n_i\times n_i},\cdots\right)\cdot\boldsymbol{S}^{-1}.$$

184

—舒尔型: 对某个酉阵 $\boldsymbol{Q}$, $\boldsymbol{A}-\lambda\boldsymbol{I}=\boldsymbol{Q}(\boldsymbol{T}-\lambda\boldsymbol{I})\boldsymbol{Q}^*$, 其中 $\boldsymbol{T}$ 是三角阵.

—实 $\boldsymbol{A}$ 的实舒尔型: 对某个实正交阵 $\boldsymbol{Q}$, $\boldsymbol{A}-\lambda\boldsymbol{I}=\boldsymbol{Q}(\boldsymbol{T}-\lambda\boldsymbol{I})\boldsymbol{Q}^{\mathrm{T}}$, 其中 $\boldsymbol{T}$ 是实准三角阵.

—应用于 ODE: 提供 $\dot{x}(t)=Ax(t)+f(t)$ 的解.

—算法: 做海森伯格约化(算法 4.6), 接着由 QR 迭代得到舒尔型(算法 4.5, 如 4.4.8 节中描述的那样执行). 可以从舒尔型计算特征向量(如 4.2.1 节中描述的那样).

—代价: 若仅要求特征值, 这个代价为 $10n^3$ flops, 若也要求 $\boldsymbol{T}$ 和 $\boldsymbol{Q}$, 则代价为 $25n^3$, 若也要求特征向量, 则代价稍稍超过 $27n^3$. 因为并非算法的所有部分都能采用 3 级 BLAS 的优点, 所以将指出代价实际上高于与之比较的矩阵乘法的

代价 $2n^3$：在 IBM RS6000/590 上计算特征值与相乘矩阵比较不是花费 $(10n^3)/(2n^3)=5$ 倍而是对 $n=100$ 它花费 23 倍，对 $n=1000$ 它花费 19 倍长时间 [10,p:162]. 在同样的机器上计算特征值和特征向量不是花费 $(27n^3)/(2n^3)=13.5$ 倍而是对 $n=100$ 它花费 41 倍对 $n=1000$ 它花费 60 倍长时间. 因此计算非对称矩阵的特征值代价是昂贵的.（对称情况便宜得多，见第 5 章）

—LAPACK：对舒尔型用 sgees，对特征值和特征向量用 sgeev，对误差界用 sgeesx 或 sgeevx.

—Matlab：对舒尔型用 Schur，对特征值和特征向量用 eig.

—利用对称性：当 $A=A^*$ 时，较好的算法在第 5 章尤其是在 5.3 节中讨论.

- 正则的 $A-\lambda B(\det(A-\lambda B)\neq 0)$

—魏尔斯特拉斯型：对某些非奇异的 P_L 和 P_R，

$$A-\lambda B=P_L\cdot\mathrm{diag}\left(\mathrm{Jordan},\begin{bmatrix}1&\lambda&&\\&\ddots&\ddots&\\&&\ddots&\lambda\\&&&1\end{bmatrix}^{n_i\times n_i}\right)P_R^{-1}.$$

—广义舒尔型：对某些酉阵 Q_L 和 Q_R, $A-\lambda B=Q_L(T_A-\lambda T_B)Q_R^*$，其中 T_A 和 T_B 是上三角阵.

185

—实 A 和 B 的广义实舒尔型：对某些实正交阵 Q_L 和 Q_R, $A-\lambda B=Q_L(T_A-\lambda T_B)Q_R^{\mathrm{T}}$,其中 T_A 是实准三角阵而 T_B 是实三角阵.

—应用于 ODEs：提供 $B\dot{x}(t)=Ax(t)+f(t)$ 的解，其中解是唯一确定的但可能非光滑地依赖于数据(脉冲响应).

—算法：海森伯格/三角形约化随即 QZ 迭代(QR 隐式地应用于 AB^{-1}).

—代价：计算 T_A 和 T_B 花费 $30n^3$. 计算 Q_L 和 Q_R 另外花费 $66n^3$. 计算全部特征向量总计花费稍少于 $69n^3$. 如前面一样 3 级 BLAS 不能被算法的所有部分使用.

—LAPACK：对 Schur 型用 sgges，对特征值用 sggev，对误差界用 sggesx 或 sggevx.

—Matlab：对特征值和特征向量用 eig.

—利用对称性：当 $A=A^*$，$B=B^*$ 且 B 正定时，利用定理 4.12 可以把问题转化为求单个对称阵的特征值. 这是用 LAPACK 程序 ssygv，sspgv(对以“压缩存储”的对称阵)和 ssbgv(对带状对称阵)来完成的.

- 奇异的 $A-\lambda B$

—克罗内克型：对某些非奇异的 P_L 和 P_R，

$$A-\lambda B=P_L\cdot\mathrm{diag}\left(\text{魏尔斯特拉斯},\begin{bmatrix}1&\lambda&&\\&\ddots&\ddots&\\&&1&\lambda\end{bmatrix}^{n_i\times(n_i+1)},\begin{bmatrix}1&&\\\lambda&\ddots&\\&\ddots&1\\&&\lambda\end{bmatrix}^{(m_i+1)\times m_i}\right)P_R^{-1}.$$

—广义上三角型：对某些酉阵 Q_L 和 Q_R，$A-\lambda B=Q_L(T_A-\lambda T_B)Q_R^*$，其中 T_A 和 T_B 是广义的上三角形，其对角块对应于克罗内克型的不同部分. 广义上三角形及算法的细节见[79,246].

—代价：算法的最普通和可靠的形式可能花费像 $O(n^4)$ 那样多，依赖于克罗内克结构的细节；这比正则的 $A-\lambda B$ 更加多. 也有一个花费稍为少些可靠的 $O(n^3)$ 的算法[27].

186

—应用于 ODEs：提供 $B\dot{x}(t)=Ax(t)+f(t)$ 的解，其中解可能是亚定的或超定的.

—软件：NETLIB/linalg/guptri.

- 矩阵多项式 $\sum_{i=0}^{d}\lambda^i A_i$[118]

—若 $A_d=I$（或 A_d 是方阵且良态足以用 $A_d^{-1}A_i$ 替代 A_i），则线性化得到标准问题

$$\begin{bmatrix} -A_{d-1} & -A_{d-2} & \cdots & \cdots & \cdots & -A_0 \\ I & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & I & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & I & 0 \end{bmatrix} - \lambda I.$$

—若 A_d 病态或奇异，线性化得到束

$$\begin{bmatrix} -A_{d-1} & -A_{d-2} & \cdots & \cdots & \cdots & -A_0 \\ I & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & I & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & I & 0 \end{bmatrix} - \lambda \begin{bmatrix} A_d & & & & \\ & I & & & \\ & & I & & \\ & & & \ddots & \\ & & & & I \end{bmatrix}.$$

4.7 第4章参考书目和其他话题

关于特征值和特征向量性质的全面的讨论见[139]. 关于特征值和特征向量扰动理论更多的细节见[161,237,52]和[10]的第4章. 定理4.7的证明见[69]. 关于魏尔斯特拉斯典范型和克罗内克典范型的讨论见[110,118]. 关于它们在系统论和控制论中的应用见[246,247,78]. 关于在计算几何、图像学和机械的CAD（计算机辅助设计）中的应用见[181,182,165]. 关于非对称特征问题并行算法的讨论见[76].

4.8 第4章问题

问题4.1（容易） 设 A 如(4.1)式中所定义的. 证明 $\det(A)=\prod_{i=1}^{b}\det(A_{ii})$ 然后证明 $\det(A-\lambda I)=\prod_{i=1}^{b}\det(A_{ii}-\lambda I)$. 由此推断 A 的特征值是 A_{11} 到 A_{bb} 的特征值

集合之并.

问题4.2(中等; Z. Bai) 假如A是正规阵, 即$AA^*=A^*A$. 证明若A也是三角阵, 则它必定是对角阵. 由此证明$n\times n$阶矩阵是正规阵当且仅当它有n个规范正交的特征向量. 提示: 证明A是正规阵当且仅当它的舒尔型是正规的.

187

问题4.3(容易; Z. Bai) 设λ和μ是A的各不相同的特征值, 设x是λ的右特征向量而y是μ的左特征向量. 证明x和y是正交的.

问题4.4(中等) 假如A有各不相同的特征值. 设$f(z)=\sum_{i=-\infty}^{+\infty}a_iz^i$是一个定义在$A$的特征值上的函数. 设$Q^*AQ=T$是$A$的舒尔型 (故$Q$是酉阵而$T$是上三角阵).

1. 证明 $f(A)=Qf(T)Q^*$. 因而为计算$f(A)$能够计算$f(T)$就足够了. 在问题的其余部分应该导出$f(T)$的一个简单的递推公式.
2. 证明 $(f(T))_{ii}=f(T_{ii})$,因此可以从T的对角元计算$f(T)$的对角元.
3. 证明 $Tf(T)=f(T)T$.
4. 从上一个结论, 证明可从$f(T)$的第$(i-1)$条上对角元和早先的上对角元计算第i条对角元. 因而, 从$f(T)$的对角元出发可计算第一上对角元, 第二上对角元等等.

问题4.5(容易) 设A是方阵. 或者应用问题4.4于A的舒尔型或者应用(4.6)式于A的若尔当型去推断$f(A)$的特征值是$f(\lambda_i)$, 其中λ_i是A的特征值. 这个结论称为**谱映射定理** (spectral mapping theorem).

这个问题用于定理6.5的证明和6.5.6节.

问题4.6(中等) 本题中将指出如何求解**西尔维斯特**(Sylvester)或**李雅普诺夫**(Lyapunov)方程$AX-XB=C$, 其中X和C是$m\times n$阶矩阵, A是$m\times m$阶矩阵而B是$n\times n$阶矩阵. 这是一个关于X的元素的mn个线性方程的方程组.

1. 给出A和B的舒尔分解, 指出如何能够把$AX-XB=C$变换到一个相似的方程组$A'Y-YB'=C'$, 其中A'和B'是上三角阵.
2. 指出如何利用类似于向后回代的过程同时求解Y的元素. 依据A和B特征值的什么条件保证方程组非奇异?
3. 指出如何变换Y去得到解X.

问题4.7(中等) 假如$T=\begin{bmatrix}A & C\\ 0 & B\end{bmatrix}$是舒尔型. 要求一个矩阵$S$使$S^{-1}TS=$

188

$\begin{bmatrix}A & 0\\ 0 & B\end{bmatrix}$. 结果是可以选择形如$\begin{bmatrix}I & R\\ 0 & I\end{bmatrix}$的$S$. 指出如何求解$R$.

问题4.8(中等; Z. Bai) 设A是$m\times n$阶矩阵而B是$n\times m$阶矩阵. 证明矩阵

$$\begin{pmatrix}AB & 0\\ B & 0\end{pmatrix}\text{和}\begin{pmatrix}0 & 0\\ B & BA\end{pmatrix}$$

相似. 由此推断 AB 的非零特征值与 BA 的非零特征值相同.

问题 4.9(中等; Z. Bai) 设 A 是具有特征值 $\lambda_1, \cdots, \lambda_n$ 的 $n \times n$ 阶矩阵. 证明

$$\sum_{i=1}^{n} |\lambda_i|^2 = \min_{\det(S) \neq 0} \|S^{-1}AS\|_F^2.$$

问题 4.10(中等; Z. Bai) 设 A 是具有特征值 $\lambda_1, \cdots, \lambda_n$ 的 $n \times n$ 阶矩阵.

1. 证明 A 可写成 $A = H + S$, 其中 $H = H^*$ 是埃尔米特阵. 而 $S = -S^*$ 是反埃尔米特阵. 根据 A 给出 H 和 S 的显式公式.

2. 证明 $\sum_{i=1}^{n} |\Re(\lambda_i)|^2 \leq \|H\|_F^2$.

3. 证明 $\sum_{i=1}^{n} |\Im(\lambda_i)|^2 \leq \|S\|_F^2$.

4. 证明 A 是正规的 $(AA^* = A^*A)$ 当且仅当 $\sum_{i=1}^{n} |\lambda_i|^2 = \|A\|_F^2$.

问题 4.11(容易) 设 λ 是一个单特征值, x 和 y 分别是右和左特征向量. 对应于 λ 定义*谱投影* P 为 $P = xy^*/(y^*x)$. 证明 P 有下列性质

1. 即使在 P 的定义中能利用 x 和 y 的任意非零标量倍, P 还是唯一的.

2. $P^2 = P$(任何满足 $P^2 = P$ 的矩阵称为*投影矩阵*(projection matrix)).

3. $AP = PA = \lambda P$(因为 P "包含" λ 的左和右不变子空间, 所以这些性质促成名称谱投影).

4. $\|P\|_2$ 是 λ 的条件数.

问题 4.12(容易; Z. Bai) 设 $A = \begin{bmatrix} a & c \\ 0 & b \end{bmatrix}$. 证明 A 的两个特征值之条件数都等于 $(1 + (\frac{c}{a-b})^2)^{1/2}$. 因此当特征值间的差 $a - b$ 相对于矩阵的非对角部分 c 小的时候, A 的条件数大.

问题 4.13(中等; Z. Bai) 设 A 是一个矩阵, x 是单位向量($\|x\|_2 = 1$), μ 是一个标量, 而 $r = Ax - \mu x$. 证明存在一个矩阵 E 具有性质 $\|E\|_F = \|r\|_2$, 使得 $A + E$ 有特征值 μ 和特征向量 x.

189

问题 4.14(中等; 程序设计) 本题中将使用一个 Matlab 程序绘出扰动的矩阵的特征值及其条件数. (它可利用 HOMEPAGE/Matlab/eigscat. m.)输入是

a = 输入矩阵,

err = 扰动的大小,

m = 计算扰动矩阵数.

输出由三个图组成, 其中每个符号是一个扰动的矩阵之特征值的位置:

"o" 标记每个未扰动的特征值位置.

"x" 标记每个扰动的特征值位置, 其中范数为 err 的一个实扰动矩阵加到 a 上.

"." 标记每个扰动的特征值位置, 其中范数为 err 的一个复扰动矩阵加到 a 上.

再打印一张 A 的特征值及其条件数的表格.

下面是一些有趣的测试例子(对你要期待的同样大的一个 m，这个 m 越大就越好，而且 m 等于几百是有益的).

(1) a = randn(5)(若没有复特征值,再试一下)

```
err = 1e-5, 1e-4, 1e-3, 1e-2, .1, .2
```

(2)
```
a = diag(ones(4,1),1); err = 1e-12, 1e-10, 1e-8
```

(3)
```
a = [[1   1e6   0      0];…
     [0   2     1e-3   0];…
     [0   0     3      10];…
     [0   0    -1      4]]
  err = 1e-8,1e-7,1e-6,1e-5,1e-4,1e-3
```

(4)
```
[q,r] = qr(randn(4,4));a = q*diag(ones(3,1),1)*q'
   err = 1e-16,1e-14,1e-12,1e-10,1e-8
```

(5)
```
a = [[1  1e3  1e6];[0  1  1e3];[0  0  1]],
   err = 1e-7,1e-6,5e-6,8e-6,1e-5,1.5e-5,2e-5
```

(6)
```
a = [[1   0   0   0   0    0];…
     [0   2   1   0   0    0];…
     [0   0   2   0   0    0];…
     [0   0   0   3   1e2  1e4];…
     [0   0   0   0   3    1e2];…
     [0   0   0   0   0    3]]
   err = 1e-10, 1e-8, 1e-6, 1e-4, 1e-3
```

190

你的任务是测试这些例子并比较特征值(所谓拟谱(pseudospectrum))占据的区域与4.3节中描述的边界．实扰动和复扰动之间的差别是什么？当扰动的 err 变为零时特征值占据的区域会发生什么情况？当 err 变为零时区域的极限大小是什么(即计算的特征值多少数位是正确的)？

问题 4.15(中等，程序设计)　本题中利用 Matlab 程序标示一个经历无位移 QR 迭代的矩阵的对角元．每个对角元的值在每次 QR 迭代后标出．每个对角元对应于一条标出的曲线(可利用 HOMEPAGE/Matlab/qrplt.m 以及下面指出的程序)．输入是

a = 输入矩阵,

m = QR 迭代次数,

而输出是标示对角元的图．

测试这个代码的例子如下(选择一个足够大的 m 使曲线或者收敛或者进入循环)：

```
a = randn(6);
b = randn(6); a = b*diag([1,2,3,4,5,6])*inv(b);
a = [[1 10]; [-1 1]]; m = 300
a = diag((1.5*ones(1,5)).\verb+^+(0:4)) +
```

```
.01 * (diag(ones(4,1),1) + diag(ones(4,1), - 1)); m = 30
```

若存在复特征值会发生什么情况?

在许多次迭代之后特征值以什么次序出现在矩阵中?

执行下列实验:假如 a 是 $n \times n$ 阶对称阵. 在 Matlab 中,设 perm = (n : −1 : 1). 这样产生一个从 n 下降到 1 的整数列表. 运行迭代 m 次. 设 a = a(perm,perm); 称这个为"非常的" a,因为它反转 a 的行和列的次序. 再运行迭代 m 次,并再构成a = a(perm,perm). 如何把这个 a 的值与原来的 a 的值比较? 你不应该让 m 太大(尝试 m = 5)否则舍入将模糊你应当看到的关系(也可见推论 5.4 和问题 5.25)

改变代码从最后的对角元值开始计算每个对角元中的误差.(只对具有实特征值的矩阵做这个工作). 画出这个误差与迭代次数相对的记录表. 你渐近地得到什么?

```
hold off
e = diag(a);
for i = 1 : m,
[q,r] = qr(a); dd = diag(sign(diag(r))); r = dd * r; q = q * dd; a = r * q; …
e = [e,diag(a)];
end
clg
plot(e','w'), grid
```

191

问题 4.16(困难,程序设计) 本题描述非线性特征问题对计算机图形学,计算几何以及机械的 CAD(计算机辅助设计)的应用;也可见[181,182,165].

设 $\boldsymbol{F} = [f_{ij}(x_1,x_2,x_3)]$是一个矩阵,其元素是三个变量 x_i 的多项式. 则 $\det(\boldsymbol{F}) = 0$ 将(一般地)定义 3 − 维空间中的一个 2 − 维曲面 S. 设 $x_1 = g_1(t), x_2 = g_2(t)$ 和 $x_3 = g_3(t)$定义一条参数为 t 的(一维)曲线 C,其中 g_i 也是多项式. 我们希望求交 $S \cap C$. 指出如何把这个问题表达为一个特征值问题(然后这个问题可以求数值解). 更一般地,说明如何求曲面 $\det(\boldsymbol{F}(x_1,\cdots,x_n)) = 0$ 和曲线$\{x_i = g_i(t), 1 \leqslant i \leqslant n\}$之交. 作为 n, $\boldsymbol{F}$ 的维数 d 和多项式f_{ij}与 g_k 次数的最大值的函数最多能有多少离散解?

编写求解这个问题的 Matlab 程序,对 $n = 3$ 个变量,通过把它转换为一个特征值问题. 它应该紧凑表达每个$f_{ij}(x_k)$和 $g_i(t)$的元素作为输入并产生一个相交点表. 例如,它可以取下列输入:

- 数组 NumTerms($1 : d, 1 : d$),其中 NumTerms(i,j) 是多项式$f_{ij}(x_1,x_2,x_3)$的项数.
- 数组 Sterms(1 : 4,1 : TotalTerms),其中 TotalTerms 是 NumTerms(. ,.)中所有元素之和. Sterms 的每列代表一个多项式中的一项: Sterms 的第一个 NumTerms(1,1)列代表f_{11}中的项,Sterms 的第二个 Numterm(2,1)列代表f_{21}中的项,等等. 由 Sterms($1 : 4, k$)代表的项是 Sterm($4,k$) · $x_1^{\mathrm{Sterm}(1,k)} \cdot x_2^{\mathrm{Sterm}(2,k)} \cdot x_3^{\mathrm{Sterm}(3,k)}$.

- 数组 $tC(1:3)$按序包含多项式 g_1，g_2 和 g_3 的次数.
- 数组 Curve$(1:tC(1)+tC(2)+tC(3)+3)$包含多项式 g_1,g_2 和 g_3 的系数，按一个多项式之后接另一个多项式，每个多项式从常数项到最高次系数排列.

你的程序也应对计算的答案计算误差界. 仅当特征问题可转化为一个应用定理4.4或4.5中误差界的问题时这件事才是可行的. 当特征问题是一个更一般的问题时，你不必提供误差界（更一般特征问题误差界的描述可见[10,237]）.

对 $n=3$ 情况编写第二个标示 S 和 C 的 Matlab 程序并标出相交点.

为运行你的代码，对输入的数据有什么限制吗？若 S 和 C 不相交时会发生什么情况？若 S 位于 C 中会发生什么情况？

至少对下面的例子运行你的代码. 你应该有能力手工求解前5个例子以检查你的代码.

192

1. $g_1=t,g_2=1+t,g_3=2+t,\boldsymbol{F}=\begin{bmatrix} x_1+x_2+x_3 & 0 \\ 0 & 3x_1+5x_2-7x_3+10 \end{bmatrix}$.

2. $g_1=t^3,g_2=1+t^3,g_3=2+t^3,\boldsymbol{F}=\begin{bmatrix} x_1+x_2+x_3 & 0 \\ 0 & 3x_1+5x_2-7x_3+10 \end{bmatrix}$.

3. $g_1=t^2,g_2=1+t^2,g_3=2+t^2,\boldsymbol{F}=\begin{bmatrix} x_1+x_2+x_3 & 0 \\ 0 & 3x_1+5x_2-7x_3+10 \end{bmatrix}$.

4. $g_1=t^2,g_2=1+t^2,g_3=2+t^2,\boldsymbol{F}=\begin{bmatrix} 1 & 0 \\ 0 & 3x_1+5x_2-7x_3+9 \end{bmatrix}$.

5. $g_1=t^2,g_2=1+t^2,g_3=2+t^2,\boldsymbol{F}=\begin{bmatrix} x_1+x_2+x_3 & 0 \\ 0 & 3x_1+5x_2-7x_3+8 \end{bmatrix}$.

6. $g_1=t^2,g_2=1+t^2,g_3=2+t^2,\boldsymbol{F}=\begin{bmatrix} x_1+x_2+x_3 & x_1 \\ x_3 & 3x_1+5x_2-7x_3+10 \end{bmatrix}$.

7. $g_1=7-3t+t^5,g_2=1+t^2+t^5,g_3=2+t^2-t^5$,

$$\boldsymbol{F}=\begin{bmatrix} x_1x_2+x_3^5 & 3-x_2^2 & 5+x_1+x_2+x_3+x_1x_2+x_1x_3+x_2x_3 \\ x_2-7x_3^5 & 1-x_1^2+x_1x_2x_3^3 & 3+x_1+3x_3-9x_2x_3 \\ 2 & 3x_1+5x_2-7x_3+8 & x_1^3-x_2^4+4x_3^5 \end{bmatrix}.$$

你应该递交

- 根据一个特征问题来表示解的数学公式.
- 算法至多2页，包含你的代码（每个高级运算的子程序名称）的路线图. 应该容易看出数学公式如何导致算法以及算法如何匹配代码.

  —至多能存在多少离散解？

—所有的计算特征值代表实际的交吗？哪些能代表？

—为使你的代码能够正确地运行，对输入设置什么限制吗？

—若 S 和 C 不相交时会发生什么情况？

—若 S 包含 C 会发生什么情况？

- 误差界的数学公式
- 计算误差界的算法至多2页，包含你的代码(每个高级运算子程序名称)的路线图. 应该容易看出数学公式如何导致算法以及如何匹配代码.
- 程序列表输出

对7个例子中的每一个，你应该递交

- 问题的原始陈述
- 产生的特征问题
- 数值解
- 标示 S 和 C；你的数值解是否匹配图表？
- 把计算解答代入确定 S 和 C 的方程中的结果. 它们(在误差范围内)是否满足？ [193]

第5章 对称特征问题和奇异值分解

5.1 概　　述

本章讨论对称特征值问题的扰动理论(5.2节)，算法(5.3节和5.4节)和应用(5.5节和在别处). 也讨论与其密切相关的SVD，因为对称阵$H=\begin{bmatrix}0 & A^T\\ A & 0\end{bmatrix}$的特征分解和$A$的SVD是非常自然地相关的(见定理3.3)，所以对称特征问题的大部分扰动定理和算法可推广到SVD.

正如第4章开始时讨论的那样，可以把对称特征问题(和SVD)的算法粗略地分成两类：*直接法*和*迭代法*. 本章只讨论直接法，它是对稠密矩阵计算全部（或经由选择的部分）特征值以及（任选的）特征向量，花费$O(n^3)$次运算. 迭代法将在第7章中讨论.

鉴于对称特征问题的算法和应用有大量的新进展，这里我们将突出三个例子：

- 基于分而治之的对称特征问题的高速算法在5.5.3节中讨论. 这是求大型稠密或带状对称阵全部特征值和全部特征向量(或一般阵的SVD)最快速和有效的算法. 它显著地快于早先的“干重活的”QR迭代算法.[1]
- 基于dqds和雅可比算法的高精度算法在5.2.1，5.2.4和5.4.3节中讨论. 这些算法比另一些类似分而治之的算法可以更精确地求微小的特征值(或奇异值). 虽然在雅可比意义下有时更慢些.
- 5.5节讨论一个“非线性”振动系统，它由一个称为*Toda流率*(Toda flow)的微分方程来描述. 它的连续解与对称特征问题QR算法的中间步是密切相关的.

仿效第4章，将继续利用振动质点-弹簧系统作为一个流动的例子说明对称特征问题的特征.

例5.1　对称特征值问题通常在分析*机械振动*(mechanical vibration)中产生. 例4.1详尽地叙述了一个这样的例子；我们将利用该例的记号，故建议读者复习一下. 为使例4.1中的问题对称，需要假定不存在阻尼，因而质点-弹簧系统运动的微分方程

---

1. 还有关于基于逆迭代(算法4.2)的一个算法的更新的工作[201,203]，它可能提供一个更快而且更准确的算法. 但是根据1997年6月的材料，理论和软件还在发展之中.

变成 $M\ddot{x}(t)=-Kx(t)$，其中 $M=\mathrm{diag}(m_1,\cdots,m_n)$，而

$$K=\begin{bmatrix} k_1+k_2 & -k_2 & & & \\ -k_2 & k_2+k_3 & -k_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-1} & k_{n-1}+k_n & -k_n \\ & & & -k_n & k_n \end{bmatrix}.$$

因为 M 非奇异，所以可将微分方程改写为 $\ddot{x}(t)=-M^{-1}Kx(t)$. 若寻找形如 $x(t)=e^{\gamma t}x(0)$ 的解，则我们得到 $e^{\gamma t}\gamma^2x(0)=-M^{-1}Ke^{\gamma t}x(0)$ 或 $M^{-1}Kx(0)=-\gamma^2x(0)$. 换言之，$-\gamma^2$ 是 $M^{-1}K$ 的一个特征值而 $x(0)$ 是一个特征向量. $M^{-1}K$ 一般不对称，但可以如下方式使它对称. 定义 $M^{1/2}=\mathrm{diag}(m_1^{1/2},\cdots,m_n^{1/2})$，在 $M^{-1}Kx(0)=-\gamma^2x(0)$ 两边用 $M^{1/2}$ 相乘，得到

$$M^{-1/2}Kx(0)=M^{-1/2}K(M^{-1/2}M^{1/2})x(0)=-\gamma^2M^{1/2}x(0)$$

或 $\hat{K}\hat{x}=-\gamma^2\hat{x}$，其中 $\hat{x}=M^{1/2}x(0)$，$\hat{K}=M^{-1/2}KM^{-1/2}$. 容易看出

$$\hat{K}=\begin{bmatrix} \frac{k_1+k_2}{m_1} & \frac{-k_2}{\sqrt{m_1m_2}} & & & \\ \frac{-k_2}{\sqrt{m_1m_2}} & \frac{k_2+k_3}{m_2} & \frac{-k_3}{\sqrt{m_2m_3}} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{-k_{n-1}}{\sqrt{m_{n-2}m_{n-1}}} & \frac{k_{n-1}+k_n}{m_{n-1}} & \frac{-k_n}{\sqrt{m_{n-1}m_n}} \\ & & & \frac{-k_n}{\sqrt{m_{n-1}m_n}} & \frac{k_n}{m_n} \end{bmatrix}$$

是对称的. 因此 $\hat{K}$ 的每个特征值 $-\gamma^2$ 是实的，且 $\hat{K}$ 的每个特征向量 $\hat{x}=M^{1/2}x(0)$ 与其他特征向量正交.

[196]

事实上，$\hat{K}$ 是三对角矩阵，利用算法 4.6，任何对称阵特别是 4.4.7 节中描述的对称阵都可化成这种特殊的三对角的形式. 5.3 节中求对称阵的特征值和特征向量的大部分算法都假定初始矩阵已经化成三对角形式.

利用 SVD，还有一种方式表示这个机械振动问题的解. 定义 $K_D=\mathrm{diag}(k_1,\cdots,k_n)$ 和 $K_D^{1/2}=\mathrm{diag}(k_1^{1/2},\cdots,k_n^{1/2})$. 则 K 可分解为 $K=BK_DB^{\mathrm{T}}$，其中

$$B=\begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & \ddots & -1 \\ & & & 1 \end{bmatrix},$$

这可通过少量计算证实. 于是

$$\begin{aligned}\hat{K} &= M^{-1/2}KM^{-1/2}\\ &= M^{-1/2}BK_DB^{\mathrm{T}}M^{-1/2}\\ &= (M^{-1/2}BK_D^{1/2})\cdot(K_D^{1/2}B^{\mathrm{T}}M^{-1/2})\\ &= (M^{-1/2}BK_D^{1/2})\cdot(M^{-1/2}BK_D^{1/2})^{\mathrm{T}}\\ &\equiv GG^{\mathrm{T}}\end{aligned} \tag{5.1}$$

所以如定理3.3中所示 $G=M^{-1/2}BK_D^{1/2}$ 的奇异值是 $\hat{K}$ 的特征值的平方根而 G 的左奇异向量是 $\hat{K}$ 的特征向量. 注意 G 仅在主对角线上和第一上对角线上具有非零元. 这种矩阵称为双对角的(bidiagonal), SVD的大部分算法首先利用4.4.7中的算法约化矩阵为双对角形式.

注意, 因为 G 是非奇异的, 所以分解 $\hat{K}=GG^{\mathrm{T}}$ 蕴含 $\hat{K}$ 是正定的. 因此 $\hat{K}$ 的特征值 $-\gamma^2$ 全部是正的. 因而 γ 是纯虚数, 而且常微分方程的解 $x(t)=e^{\gamma t}x(0)$ 以频率 $|\gamma|$ 振动.

振动质点-弹簧系统的Matlab解法见HOMEPAGE/Matlab/massspring.m. 一个类似的物理系统振动的Matlab动画, 输入demo然后单击continue/fun-extras/miscellaneous/bending. ◇

5.2 扰动理论

197

假定 A 对称, 具有特征值 $\alpha_1\geq\cdots\geq\alpha_n$ 和对应的单位特征向量 $q_1,\cdots,q_n$. 假定 E 也对称, 且设 $\hat{A}=A+E$ 有扰动的特征值 $\hat{\alpha}_1\geq\cdots\geq\hat{\alpha}_n$ 和对应的扰动的特征向量 $\hat{q}_1,\cdots,\hat{q}_n$. 本节的主要目标是根据 E 的"大小"界定特征值 α_i 和 $\hat{\alpha}_i$ 之间的差以及特征向量 q_i 和 $\hat{q}_i$ 之间的差. 除了讨论"相对"扰动理论的5.2.1节之外, 大部分的界将利用 $\|E\|_2$ 作为 E 的大小.

第4章中已导出特征值的第一个扰动界, 那里已证明了推论4.1: 设 A 对称具有特征值 $\alpha_1\geq\cdots\geq\alpha_n$. 设 $A+E$ 对称具有特征值 $\hat{\alpha}_1\geq\cdots\geq\hat{\alpha}_n$. 若 α_i 是单重的, 则 $|\alpha_i-\hat{\alpha}_i|\leq\|E\|_2+O(\|E\|_2^2)$.

这个结果是弱的, 因为它假设 α_i 重数为1, 而且它仅对充分小的 $\|E\|_2$ 是有用的. 下列定理同时去掉这两个弱点.

定理5.1 外尔(Weyl). 设 A 和 E 是 $n\times n$ 阶对称阵. 设 $\alpha_1\geq\cdots\geq\alpha_n$ 是 A 的特征值而 $\hat{\alpha}_1\geq\cdots\geq\hat{\alpha}_n$ 是 $\hat{A}=A+E$ 的特征值. 则 $|\alpha_i-\hat{\alpha}_i|\leq\|E\|_2$.

推论5.1 设 G 和 F 是(同样大小的)任意矩阵, 其中 $\sigma_1\geq\cdots\geq\sigma_n$ 是 G 的奇异值而 $\sigma'_1\geq\cdots\geq\sigma'_n$ 是 $G+F$ 的奇异值, 则 $|\sigma_i-\sigma'_i|\leq\|F\|_2$.

可以利用外尔定理通过任何诸如QR迭代那样向后稳定的算法得到计算的特征值的误差界: 这样的一个算法计算出的特征值 $\hat{\alpha}_i$ 是 $\hat{A}=A+E$ 的精确的特征值, 其中

$\|E\|_2 = O(\varepsilon)\|A\|_2$. 所以，它们的误差能以 $|\alpha_i - \hat{\alpha}_i| \leqslant \|E\|_2 = O(\varepsilon)\|A\|_2 = O(\varepsilon)$ $\max_j |\alpha_j|$为界. 特别地对大的特征值(那些 α_i 数量上接近$\|A\|_2$)这是一个非常满意的误差界，因为它们将算得其大多数的正确数字. 小的特征值($|\alpha_i| \ll \|A\|_2$) 可能有较少的正确数字. (见5.2.1 节)

我们将利用另一些有用的经典结果：柯朗 - 费希尔极小极大定理证明外尔定理. 为叙述这个定理需要引入瑞利商(Rayleigh quotient)，它也将在几个算法，诸如算法 5.1 中扮演重要的角色.

定义 5.1 对称矩阵 A 和非零向量 u 的瑞利商是 $\rho(u,A) \equiv (u^{\mathrm{T}}Au)/(u^{\mathrm{T}}u)$.

下面是 $\rho(u,A)$的一些简单而重要的性质. 首先，对任何非零的标量 γ, $\rho(\gamma u,A) = \rho(u,A)$. 其次，若 $Aq_i = \alpha_i q_i$，则 $\rho(q_i,A) = \alpha_i$. 更一般地，假如 $Q^{\mathrm{T}}AQ = \Lambda = \mathrm{diag}(\alpha_i)$是 A 的特征分解，其中 $Q = [q_1, \cdots, q_n]$. u 按特征向量 q_i 构成的基展开如下：$u = Q(Q^{\mathrm{T}}u) \equiv Q\xi = \sum_i q_i \xi_i$. 于是可记

$$\rho(u,A) = \frac{\xi^{\mathrm{T}}Q^{\mathrm{T}}AQ\xi}{\xi^{\mathrm{T}}Q^{\mathrm{T}}Q\xi} = \frac{\xi^{\mathrm{T}}\Lambda\xi}{\xi^{\mathrm{T}}\xi} = \frac{\sum_i \alpha_i \xi_i^2}{\sum_i \xi_i^2}.$$

换言之，$\rho(u,A)$是 A 的特征值的加权平均. 它的最大值 $\max_{u\neq 0}\rho(u,A)$出现在 $u = q_1 (\xi = e_1)$而且等于$\rho(q_1,A) = \alpha_1$. 它的最小值 $\min_{u\neq 0}\rho(u,A)$出现在 $u = q_n(\xi = e_n)$而且等于$\rho(q_n,A) = \alpha_n$. 合计这些事实推出 198

$$\max_{u\neq 0}|\rho(u,A)| = \max(|\alpha_1|, |\alpha_n|) = \|A\|_2. \tag{5.2}$$

定理 5.2 柯朗 - 费希尔极小极大定理. 设 $\alpha_1 \geqslant \cdots \geqslant \alpha_n$ 是对称阵 A 的特征值而 $q_1, \cdots, q_n$ 是对应的单位特征向量.

$$\max_{\mathbf{R}^j} \min_{0\neq r\in \mathbf{R}^j} \rho(r,A) = \alpha_j = \min_{\mathbf{S}^{n-j+1}} \max_{0\neq s\in \mathbf{S}^{n-j+1}} \rho(s,A).$$

α_j 的第 1 个表达式中的极大值是取遍$\mathbb{R}^n$ 的所有 j 维子空间 $\mathbf{R}^j$，而随后的极小值是取遍子空间中的所有非零向量 r. 极大值是对 $\mathbf{R}^j = \mathrm{span}(q_1, q_2, \cdots, q_j)$达到的，而一个极小的 r 是 $r = q_j$.

α_j 的第 2 个表达式中的极小值是取遍$\mathbb{R}^n$ 的所有$(n-j+1)$维子空间 $\mathbf{S}^{n-j+1}$，而随后的极大值是取遍子空间中的所有非零向量 s. 极小值是对 $\mathbf{S}^{n-j+1} = \mathrm{span}(q_j, q_{j+1}, \cdots, q_n)$达到的，而一个极大的 s 是 $s = q_j$.

例 5.2 设 $j=1$，故 α_j 是最大的特征值. 给定 $\mathbf{R}^1$, $\rho(r,A)$对所有非零的 $r \in \mathbf{R}^1$ 是相同的，因为所有这样的 r 彼此是标量倍. 于是 α_1 的第 1 个表达式简化为 $\alpha_1 = \max_{r\neq 0}\rho(r,A)$. 类似地，因为 $n-j+1 = n$，所以仅有的子空间 $\mathbf{S}^{n-j+1}$是全空间$\mathbb{R}^n$. 因此 α_1 的第 2 个表达式也简化为 $\max_{s\neq 0}\rho(s,A)$.

对最小的特征值我们可类似地证明定理能简化为下列表达式：$\alpha_n = \min_{r\neq 0}\rho(r,A)$. ◇

柯朗 - 费希尔定理的证明 选择指定维数的任意子空间 $\mathbf{R}^j$ 和 $\mathbf{S}^{n-j+1}$. 因为它们的维

数之和$j+(n-j+1)=n+1$超过n，所以必存在一个非零向量$x_{\mathrm{RS}}\in\mathbf{R}^j\cap\mathbf{S}^{n-j+1}$. 因此

$$\min_{0\neq r\in\mathbf{R}^j}\rho(r,A)\leqslant\rho(x_{\mathrm{RS}},A)\leqslant\max_{0\neq s\in\mathbf{S}^{n-j+1}}\rho(s,A).$$

现在选择$\hat{\mathbf{R}}^j$极大化左边的表达式，并选择$\hat{\mathbf{S}}^{n-j+1}$极小化右边的表达式. 则

$$\begin{aligned}\max_{\mathbf{R}^j}\min_{0\neq r\in\mathbf{R}^j}\rho(r,A)&=\min_{0\neq r\in\hat{\mathbf{R}}^j}\rho(r,A)\\&\leqslant\rho(x_{\hat{\mathbf{R}}\hat{\mathbf{S}}},A)\\&\leqslant\max_{0\neq s\subset\hat{\mathbf{S}}^{n-j+1}}\rho(s,A)\\&=\min_{\mathbf{S}^{n-j+1}}\max_{0\neq s\in\mathbf{S}^{n-j+1}}\rho(s,A).\end{aligned}\tag{5.3}$$

199

为观察所有这些不等式实际上是等式，我们提交特别的$\boldsymbol{R}^j$和$\boldsymbol{S}^{n-j+1}$使下界等于上界. 首先选择$\underline{\mathbf{R}}^j=\mathrm{span}(\boldsymbol{q}_1,\cdots,\boldsymbol{q}_j)$使得

$$\begin{aligned}\max_{\mathbf{R}^j}\min_{0\neq r\in\mathbf{R}^j}\rho(r,A)&\geqslant\min_{0\neq r\in\underline{\mathbf{R}}^j}\rho(\boldsymbol{r},\boldsymbol{A})\\&=\min_{0\neq r=\sum_{i\leqslant j}\xi_i q_i}\rho(\boldsymbol{r},\boldsymbol{A})\\&=\min_{\text{some }\xi_i\neq0}\frac{\sum_{i\leqslant j}\xi_i^2\alpha_i}{\sum_{i\leqslant j}\xi_i^2}=\alpha_j.\end{aligned}$$

下面选择$\overline{\mathbf{S}}^{n-j+1}=\mathrm{span}(\boldsymbol{q}_j,\cdots,\boldsymbol{q}_n)$使得

$$\begin{aligned}\min_{\mathbf{S}^{n-j+1}}\max_{0\neq s\in\mathbf{S}^{n-j+1}}\rho(\boldsymbol{s},\boldsymbol{A})&\leqslant\max_{0\neq s\in\overline{\mathbf{S}}^{n-j+1}}\rho(\boldsymbol{s},\boldsymbol{A})\\&=\max_{0\neq s=\sum_{i\geqslant j}\xi_i q_i}\rho(\boldsymbol{s},\boldsymbol{A})\\&=\max_{\text{some }\xi\neq0}\frac{\sum_{i\geqslant j}\xi_i^2\alpha_i}{\sum_{i\geqslant j}\xi_i^2}=\alpha_j.\end{aligned}$$

于是，下界和上界夹在α_j下面和α_j上面之间，故它们必须如要求的那样都等于α_j. □

例5.3　图5-1(见彩插)对3×3阶矩阵从图形上阐明这个定理. 因为$\rho(\boldsymbol{u}/\|\boldsymbol{u}\|_2,\ \boldsymbol{A})=\rho(\boldsymbol{u},\boldsymbol{A})$，所以可以认为$\rho(\boldsymbol{u},\boldsymbol{A})$为单位球面$\|\boldsymbol{u}\|_2=1$上的一个函数. 图5-1对$\boldsymbol{A}=\mathrm{diag}(1,0.25,0)$指出在单位球面上这个函数的周线图. 对这个简单的矩阵$\boldsymbol{q}_i=\boldsymbol{e}_i$, $\boldsymbol{e}_i$为单位阵的第i列. 因为$\rho(\boldsymbol{u},\boldsymbol{A})=\rho(-\boldsymbol{u},\boldsymbol{A})$，所以图形关于原点对称. 靠近$\pm\boldsymbol{q}_1$的小红圆围绕整体极大值$\rho(\pm\boldsymbol{q}_1,\boldsymbol{A})=1$，而靠近$\pm\boldsymbol{q}_3$的小绿圆围绕整体极小值$\rho(\pm\boldsymbol{q}_3,\boldsymbol{A})=0$. 两个大圆是第2个特征值$\rho(\boldsymbol{u},\boldsymbol{A})=0.25$的周线. 在由最大的圆限定的两个狭窄的(绿色)“苹果切片”内部$\rho(\boldsymbol{u},\boldsymbol{A})<0.25$，而在宽的(红色)苹果切片内部$\rho(\boldsymbol{u},\boldsymbol{A})>0.25$.

根据这个图形来解释极小极大定理. 选择空间$\mathbf{R}^2$等价于选择大圆C；C上的每个点位于$\mathbf{R}^2$内部，而$\mathbf{R}^2$由C中向量的所有标量倍组成. 于是$\min_{0\neq r\in\mathbf{R}^j}\rho(\boldsymbol{r},\boldsymbol{A})=\min_{r\in C}\rho(\boldsymbol{r},\boldsymbol{A})$. 为计算$\min_{r\in C}\rho(\boldsymbol{r},\boldsymbol{A})$有四种情况需要考虑.

1. C不通过图5-1中两个大圆的交点$\pm q_2$. 因而C显然必定同时与狭窄的绿色苹

果切片和宽的红色苹果切片相交，故 $\min_{r\in C}\rho(\boldsymbol{r},\boldsymbol{A})<0.25$.

2. C 通过两个交点 $\pm q_2$ 并且另外的位于狭窄的绿色苹果切片之中. 因而 $\min_{r\in C}\rho(\boldsymbol{r},\boldsymbol{A})<0.25$.

200

3. C 通过两个交点 $\pm q_2$ 并且另外的位于宽的红色苹果切片之中. 因而在 $\boldsymbol{r}=\pm\boldsymbol{q}_2$ 时达到 $\min_{r\in C}\rho(\boldsymbol{r},\boldsymbol{A})=0.25$.

4. C 与两个大圆之一相同. 因而对一切 $r\in C,\rho(\boldsymbol{r},\boldsymbol{A})=0.25$.

极小极大定理断言取遍所有选择的大圆 $C,\alpha_2=0.25$ 是 $\min_{r\in C}\rho(\boldsymbol{r},\boldsymbol{A})$的极大值. 这个极大值在上面的情况 3 和情况 4 中达到. 特别地，C 对分宽的红色苹果切片(情况 3)，$\mathbf{R}^2=\mathrm{span}(q_1,q_2)$.

对任一个 3×3 阶对称阵画类似图 5-1 中的那些周线图的软件可在 HOMEPAGE/Matlab/RayleighContour. m 中找到. ◇

最后，我们给出外尔定理的证明.

$$\begin{aligned}\hat{\alpha}_j &= \min_{S^{n-j+1}}\max_{0\neq u\in S^{n-j+1}}\frac{\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{A}+\boldsymbol{E})\boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}} \quad \text{由极小极大定理}\\ &= \min_{S^{n-j+1}}\max_{0\neq u\in S^{n-j+1}}\left(\frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}}+\frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{E}\boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}}\right)\\ &\leqslant \min_{S^{n-j+1}}\max_{0\neq u\in S^{n-j+1}}\left(\frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}}+\|\boldsymbol{E}\|_2\right) \quad \text{由(5.2)式}\\ &= \alpha_j+\|\boldsymbol{E}\|_2 \quad \text{再次由极小极大定理.}\end{aligned}$$

交换 $\boldsymbol{A}$ 和 $\boldsymbol{A}+\boldsymbol{E}$ 的角色，也可得到 $\alpha_j\leqslant\hat{\alpha}_j+\|\boldsymbol{E}\|_2$. 合并这两个不等式完成外尔定理的证明. □

在后面 5.3.4 节中论证对分算法所需的一个定理是与柯朗 - 费希尔极小极大定理密切相关的西尔维斯特(Sylvester)惯性定理.

201

定义 5.2 对称阵 $\boldsymbol{A}$ 的惯性是三元整数组 $\mathrm{Inertia}(\boldsymbol{A})\equiv(\nu,\zeta,\pi)$，其中 ν 是 $\boldsymbol{A}$ 的负特征值个数，ζ 是 $\boldsymbol{A}$ 的零特征值个数，而 π 是 $\boldsymbol{A}$ 的正特征值个数.

若 $\boldsymbol{X}$ 正交，则 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}$ 和 $\boldsymbol{A}$ 相似，因此有相同的特征值. 仅当 $\boldsymbol{X}$ 为非奇异时，称 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}$ 和 $\boldsymbol{A}$ 是同余 (congruent) 的. 此时 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}$ 一般没有像 $\boldsymbol{A}$ 那样相同的特征值，但下面的定理告诉我们两组特征值将至少有相同的符号.

定理 5.3 西尔维斯特惯性定理. 设 $\boldsymbol{A}$ 对称且 $\boldsymbol{X}$ 非奇异. 则 $\boldsymbol{A}$ 和 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}$ 有相同的惯性.

证明 设 n 是 $\boldsymbol{A}$ 的维数. 现在假定 $\boldsymbol{A}$ 有 ν 个负的特征值而 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}$ 有 $\nu'<\nu$ 个负的特征值. 为证明这不会发生，我们将找一个矛盾. 设 N 是 $\boldsymbol{A}$ 的对应的 ν 维负特征空间；即 N 是由 $\boldsymbol{A}$ 的 ν 个负特征值的特征向量张成的. 这意味着对任何非零的 $x\in N$，$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}<0$. 设 P 是 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}$ 的$(n-\nu')$-维非负的特征空间；这意味着对任何非零的 $x\in P$，$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}\boldsymbol{x}\geqslant 0$. 因为 $\boldsymbol{X}$ 非奇异，所以空间 XP 也是 $n-\nu'$维的. 因为$\dim(N)+\dim(XP)=\nu+n-\nu'>n$，所以空间 N 和 XP 在它们的交中必含一个非零的向量 $\boldsymbol{x}$. 但是因为

$x \in N$，$0 > x^T Ax$ 且因为 $x \in XP$，$0 \leqslant x^T Ax$，这是一个矛盾. 所以 $v \geqslant v'$，交换 A 和 $X^T AX$ 的角色，也可得到 $v' \leqslant v$，因而 $v = v'$，即 A 和 $X^T AX$ 有相同个数的负特征值. 类似的论证指出它们有相同个数的正特征值. 因此，它们也必有相同个数的零特征值. □

现在考虑把 A 扰动为 $A+E$ 时 A 的特征向量会怎样改变. 为叙述界需要定义谱中的间隙(gap).

定义 5.3　设 A 有特征值 $\alpha_1 \geqslant \cdots \geqslant \alpha_n$. 则特征值 α_i 与谱的其余特征值之间的间隙定义为 $\text{gap}(i,A) = \min_{j \neq i} |\alpha_j - \alpha_i|$. 若从上下文知道 A 的话也可记作 $\text{gap}(i)$.

基本的结果是特征向量的敏感性依赖于它对应的特征值的间隙：一个小的间隙暗示一个敏感的特征向量.

例 5.4　设 $A = \begin{bmatrix} 1+g & \\ & 1 \end{bmatrix}$ 和 $A+E = \begin{bmatrix} 1+g & \varepsilon \\ \varepsilon & 1 \end{bmatrix}$. 其中 $0 < \varepsilon < g$. 因此 $\text{gap}(i,A) = g \approx \text{gap}(i,A+E), i=1,2$. A 的特征向量正好是 $q_1 = e_1$ 和 $q_2 = e_2$. 少量的计算得出 $A+E$ 的特征向量为

202

$$\hat{q}_1 = \beta \cdot \begin{bmatrix} 1 + \sqrt{1 + \left(\frac{2\varepsilon}{g}\right)^2} \\ \frac{2\varepsilon}{g} \end{bmatrix} \approx \begin{bmatrix} 1 \\ \frac{\varepsilon}{g} \end{bmatrix},$$

$$\hat{q}_2 = \beta \cdot \begin{bmatrix} -\frac{2\varepsilon}{g} \\ 1 + \sqrt{1 + \left(\frac{2\varepsilon}{g}\right)^2} \end{bmatrix} \approx \begin{bmatrix} -\frac{\varepsilon}{g} \\ 1 \end{bmatrix},$$

其中 $\beta \approx 1/2$ 是一个规格化因子. 可以看出扰动的向量 $\hat{q}_i$ 和未扰动的向量 q_i 之间的夹角就一阶的 ε 而言等于 ε/g. 故夹角与间隙 g 的倒数成比例.

一般情况本质上与刚才分析的 2×2 阶矩阵情况相同. ◇

定理 5.4　设 $A = Q\Lambda Q^T = Q\,\text{diag}(\alpha_i)Q^T$ 是 A 的特征分解. 设 $A+E = \hat{A} = \hat{Q}\hat{\Lambda}\hat{Q}^T$ 是扰动的特征分解. 记 $Q = [q_1, \cdots, q_n]$ 和 $\hat{Q} = [\hat{q}_1, \cdots, \hat{q}_n]$，其中 q_i 和 $\hat{q}_i$ 分别是未扰动和扰动的特征向量. 设 θ 表示 q_i 和 $\hat{q}_i$ 之间的锐角. 则

$$\frac{1}{2}\sin 2\theta \leqslant \frac{\|E\|_2}{\text{gap}(i,A)}, \text{倘若 } \text{gap}(i,A) > 0.$$

类似地

$$\frac{1}{2}\sin 2\theta \leqslant \frac{\|E\|_2}{\text{gap}(i,A+E)}, \text{倘若 } \text{gap}(i,A+E) > 0.$$

注意当 $\theta \ll 1$ 时，$(1/2)\sin 2\theta \approx \sin\theta \approx \theta$.

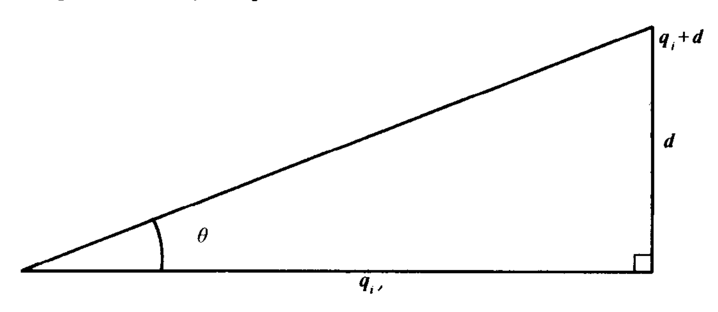
根据 $\text{gap}(i,A+E)$ 和 $\text{gap}(i,A)$ 表述的界之吸引力是我们经常只知道 $A+E$ 的特

征值,因为它们是我们所使用的特征值算法特有的输出结果.此时,计算 gap($i,\boldsymbol{A}+\boldsymbol{E}$)是直接的,但是 gap($i,\boldsymbol{A}$)只能够估计.

当第 1 个上界超过 1/2 时,即$\|\boldsymbol{E}\|_2 \geq \text{gap}(i,\boldsymbol{A})/2$,界化为 $\sin 2\theta \leq 1$,这个界不提供关于 θ 的信息.下面是在这种情况下为什么不能界定 θ 的说明:若 $\boldsymbol{E}$ 是这样大,尽管 $\boldsymbol{A}+\boldsymbol{E}$在 $\hat{\alpha}_i$ 有重特征值,$\boldsymbol{A}+\boldsymbol{E}$ 的特征值 $\hat{\alpha}_i$ 可充分远离 α_i.例如,考虑 $\boldsymbol{A}=\text{diag}(2,0)$ 和 $\boldsymbol{A}+\boldsymbol{E}=\boldsymbol{I}$.可是这样的一个 $\boldsymbol{A}+\boldsymbol{E}$ 没有唯一的特征向量 $\boldsymbol{q}_i$;实际上任意的向量都可作为 $\boldsymbol{A}+\boldsymbol{E}=\boldsymbol{I}$ 的特征向量.因此试图界定 θ 毫无意义.同样的理由适用于第 2 个上界超过 1/2 的情况.

证明 证明第 1 个上界就足够了,因为把 $\boldsymbol{A}+\boldsymbol{E}$ 看作为未扰动矩阵而把 $\boldsymbol{A}=(\boldsymbol{A}+\boldsymbol{E})-\boldsymbol{E}$ 作为扰动矩阵可得第 2 个上界.

设 $\boldsymbol{q}_i+\boldsymbol{d}$ 是 $\boldsymbol{A}+\boldsymbol{E}$ 的特征向量. 为使 $\boldsymbol{d}$ 唯一,如下面所示,强加它正交于 $\boldsymbol{q}_i$(记作 $\boldsymbol{d}\perp\boldsymbol{q}_i$). 注意,这意味着 $\boldsymbol{q}_i+\boldsymbol{d}$ 不是一个单位向量,因此 $\hat{\boldsymbol{q}}_i=(\boldsymbol{q}_i+\boldsymbol{d})/\|\boldsymbol{q}_i+\boldsymbol{d}\|_2$.于是 $\tan\theta=\|\boldsymbol{d}\|_2$, $\sec\theta=\|\boldsymbol{q}_i+\boldsymbol{d}\|_2$. [203]



现记 $(\boldsymbol{A}+\boldsymbol{E})\hat{\boldsymbol{Q}}=\hat{\boldsymbol{Q}}\hat{\boldsymbol{\Lambda}}$ 的第 i 列为

$$(\boldsymbol{A}+\boldsymbol{E})(\boldsymbol{q}_i+\boldsymbol{d})=\hat{\alpha}_i(\boldsymbol{q}_i+\boldsymbol{d}), \tag{5.4}$$

这里已经用$\|\boldsymbol{q}_i+\boldsymbol{d}\|_2$ 乘(5.4)式的两边. 定义 $\eta=\hat{\alpha}_i-\alpha_i$. 从(5.4)的两边减去 $\boldsymbol{A}\boldsymbol{q}_i=\alpha_2\boldsymbol{q}_i$ 并重新排列得到

$$(\boldsymbol{A}-\alpha_i\boldsymbol{I})\boldsymbol{d}=(\eta\boldsymbol{I}-\boldsymbol{E})(\boldsymbol{q}_i+\boldsymbol{d}). \tag{5.5}$$

因为 $\boldsymbol{q}_i^{\mathrm{T}}(\boldsymbol{A}-\alpha_i\boldsymbol{I})=0$,所以(5.5)的两边正交于 $\boldsymbol{q}_i$. 这就可记 $z\equiv(\eta\boldsymbol{I}-\boldsymbol{E})(\boldsymbol{q}_i+\boldsymbol{d})=\sum_{j\neq i}\zeta_j\boldsymbol{q}_j$. 和 $\boldsymbol{d}\equiv\sum_{j\neq i}\delta_j\boldsymbol{q}_j$. 因为$(\boldsymbol{A}-\alpha_i\boldsymbol{I})q_j=(\alpha_j-\alpha_i)\boldsymbol{q}_j$, 所以可记

$$(\boldsymbol{A}-\alpha_i\boldsymbol{I})\boldsymbol{d}=\sum_{j\neq i}(\alpha_j-\alpha_i)\delta_j\boldsymbol{q}_j=\sum_{j\neq i}\zeta_j\boldsymbol{q}_j=(\eta\boldsymbol{I}-\boldsymbol{E})(\boldsymbol{q}_i+\boldsymbol{d})$$

或

$$\boldsymbol{d}=\sum_{j\neq i}\delta_j q_j=\sum_{j\neq i}\frac{\zeta_j}{\alpha_j-\alpha_i}\boldsymbol{q}_j.$$

因而

$$\tan\theta=\|\boldsymbol{d}\|_2$$

$$
\begin{aligned}
&= \left\| \sum_{j\neq i} \frac{\zeta_j}{\alpha_j-\alpha_i} \boldsymbol{q}_j \right\|_2 \\
&= \left(\sum_{j\neq i} \left(\frac{\zeta_j}{\alpha_j-\alpha_i} \right)^2 \right)^{1/2} \quad \text{因为 } q_j \text{ 是规范正交的} \\
&\leqslant \frac{1}{\mathrm{gap}(i,\boldsymbol{A})} \left(\sum_{j\neq i} \zeta_j^2 \right)^{1/2} \quad \text{因为 gap}(i,A)\text{是最小的分母} \\
&= \frac{\|z\|_2}{\mathrm{gap}(i,\boldsymbol{A})}.
\end{aligned}
$$

204

若我们使用外尔定理和三角不等式去界定 $\|z\|_2 \leqslant (\|\boldsymbol{E}\|_2 + |\eta|) \cdot \|\boldsymbol{q}_i+\boldsymbol{d}\|_2 \leqslant 2\|\boldsymbol{E}\|_2\sec\theta$，则可得出 $\sin\theta \leqslant 2\|\boldsymbol{E}\|_2/\mathrm{gap}(i,\boldsymbol{A})$.

但是通过更仔细地界定 $\|z\|_2 = \|(\eta\boldsymbol{I}-\boldsymbol{E})(\boldsymbol{q}_i+\boldsymbol{d})\|_2$ 可以做得比这个界稍为好一些：用 $\boldsymbol{q}_i^T$ 乘(5.4)的两边，消项并重新安排得到 $\eta = \boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{E}(\boldsymbol{q}_i+\boldsymbol{d})$. 记

$$
\begin{aligned}
z &= (\boldsymbol{q}_i+\boldsymbol{d})\eta - \boldsymbol{E}(\boldsymbol{q}_i+\boldsymbol{d}) = (\boldsymbol{q}_i+\boldsymbol{d})\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{E}(\boldsymbol{q}_i+\boldsymbol{d}) - \boldsymbol{E}(\boldsymbol{q}_i+\boldsymbol{d}) \\
&= ((\boldsymbol{q}_i+\boldsymbol{d})\boldsymbol{q}_i^{\mathrm{T}} - \boldsymbol{I})\boldsymbol{E}(\boldsymbol{q}_i+\boldsymbol{d}),
\end{aligned}
$$

因而 $\|z\|_2 \leqslant \|(\boldsymbol{q}_i+\boldsymbol{d})\boldsymbol{q}_i^{\mathrm{T}}-\boldsymbol{I}\| \cdot \|\boldsymbol{E}\|_2 \cdot \|\boldsymbol{q}_i+\boldsymbol{d}\|$我们要求 $\|(\boldsymbol{q}_i+\boldsymbol{d})\boldsymbol{q}_i^{\mathrm{T}}-\boldsymbol{I}\|_2 = \|\boldsymbol{q}_i+\boldsymbol{d}\|_2$（见问题5.7）. 从而 $\|z\|_2 \leqslant \|\boldsymbol{q}_i+\boldsymbol{d}\|_2^2 \cdot \|\boldsymbol{E}\|_2$，故

$$
\tan\theta \leqslant \frac{\|z\|_2}{\mathrm{gap}(i,\boldsymbol{A})} \leqslant \frac{\|\boldsymbol{q}_i+\boldsymbol{d}\|_2^2\|\boldsymbol{E}\|_2}{\mathrm{gap}(i,\boldsymbol{A})} = \frac{\sec^2\theta \cdot \|\boldsymbol{E}\|_2}{\mathrm{gap}(i,\boldsymbol{A})}
$$

或如要求的那样

$$
\frac{\|\boldsymbol{E}\|_2}{\mathrm{gap}(i,\boldsymbol{A})} \geqslant \frac{\tan\theta}{\sec^2\theta} = \sin\theta\cos\theta = \frac{1}{2}\sin 2\theta. \qquad \square
$$

对奇异向量可以证明一个类似的定理（见问题5.8）.

瑞利商具有其他一些好的性质. 下列定理告诉我们在自然的意义下瑞利商是特征值的一个"最佳逼近". 这是5.3.2节中瑞利商迭代以及第7章中迭代算法的基础. 它也可用于任何估计方法，不只是这里讨论的算法得到的近似特征对的精度.

定理5.5 *设 $\boldsymbol{A}$ 对称，$\boldsymbol{x}$ 是单位向量而 β 是标量. 则 $\boldsymbol{A}$ 有满足 $|\alpha_i-\beta| \leqslant \|\boldsymbol{A}x-\beta x\|_2$ 的特征对 $\boldsymbol{A}\boldsymbol{q}_i = \alpha_i\boldsymbol{q}_i$. 给定 x，选择 $\beta = \rho(\boldsymbol{x},\boldsymbol{A})$ 使 $\|\boldsymbol{A}x-\beta\boldsymbol{x}\|_2$ 达到极小.*

再多用一些关于 $\boldsymbol{A}$ 的谱的信息，可以得到紧凑的界. 设 $r = \boldsymbol{A}x - \rho(\boldsymbol{x},\boldsymbol{A})\boldsymbol{x}$. α_i 是 $\boldsymbol{A}$ 最靠近于 $\rho(x,\boldsymbol{A})$ 的特征值. 设 $\mathrm{gap}' \equiv \min_{j\neq i}|\alpha_j-\rho(\boldsymbol{x},\boldsymbol{A})|$；这是早先定义的间隙的一个变形. 设 θ 是 $\boldsymbol{x}$ 和 $\boldsymbol{q}_i$ 之间的锐角. 则

$$
\sin\theta \leqslant \frac{\|\boldsymbol{r}\|_2}{\mathrm{gap}'} \tag{5.6}
$$

且

$$
|\alpha_i - \rho(\boldsymbol{x},\boldsymbol{A})| \leqslant \frac{\|\boldsymbol{r}\|_2^2}{\mathrm{gap}'}. \tag{5.7}
$$

这个结果推广到一组特征值的情况见定理7.1.

注意在(5.7)式中瑞利商$\rho(\boldsymbol{x},\boldsymbol{A})$与一个特征值$\boldsymbol{\alpha}_i$之差与残差范数$\|\boldsymbol{r}\|_2$的平方成比例，这个高精度是5.3.2节的瑞利商迭代立方收敛性的基础. [205]

证明 只证明第一个结论，其余作为本章末尾的问题5.9和问题5.10.

若β是$\boldsymbol{A}$的一个特征值则直接可得结论. 故假定改为$\boldsymbol{A}-\beta\boldsymbol{I}$非奇异. 则$\boldsymbol{x}=(\boldsymbol{A}-\beta\boldsymbol{I})^{-1}(\boldsymbol{A}-\beta\boldsymbol{I})\boldsymbol{x}$且

$$1=\|\boldsymbol{x}\|_2\leqslant\|(\boldsymbol{A}-\beta\boldsymbol{I})^{-1}\|_2\cdot\|(\boldsymbol{A}-\beta\boldsymbol{I})\boldsymbol{x}\|_2.$$

记$\boldsymbol{A}$的特征分解为$\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}}=\boldsymbol{Q}\mathrm{diag}(\alpha_1,\cdots,\alpha_n)\boldsymbol{Q}^{\mathrm{T}}$，我们得到

$$\|(\boldsymbol{A}-\beta\boldsymbol{I})^{-1}\|_2=\|\boldsymbol{Q}(\boldsymbol{\Lambda}-\beta\boldsymbol{I})^{-1}\boldsymbol{Q}^{\mathrm{T}}\|_2=\|(\boldsymbol{\Lambda}-\beta\boldsymbol{I})^{-1}\|_2=1/\min_i|\alpha_i-\beta|,$$

故如要求的那样$\min_i|\alpha_i-\beta|\leqslant\|(\boldsymbol{A}-\beta\boldsymbol{I})x\|_2$.

为证明$\beta=\rho(\boldsymbol{x},\boldsymbol{A})$使$\|\boldsymbol{A}\boldsymbol{x}-\beta\boldsymbol{x}\|_2$达到极小，我们将证明$\boldsymbol{x}$正交于$\boldsymbol{A}\boldsymbol{x}-\rho(\boldsymbol{x},\boldsymbol{A})\boldsymbol{x}$，然后应用毕达哥拉斯定理于正交向量之和

$$\boldsymbol{A}\boldsymbol{x}-\boldsymbol{\beta}\boldsymbol{x}=[\boldsymbol{A}\boldsymbol{x}-\rho(\boldsymbol{x},\boldsymbol{A})\boldsymbol{x}]+[(\rho(\boldsymbol{x},\boldsymbol{A})-\beta)\boldsymbol{x}]$$

得到

$$\begin{aligned}\|\boldsymbol{A}\boldsymbol{x}-\beta\boldsymbol{x}\|_2^2&=\|\boldsymbol{A}\boldsymbol{x}-\rho(\boldsymbol{x},\boldsymbol{A})\boldsymbol{x}\|_2^2+\|(\rho(\boldsymbol{x},\boldsymbol{A})-\beta)\boldsymbol{x}\|_2^2\\&\geqslant\|\boldsymbol{A}\boldsymbol{x}-\rho(\boldsymbol{x},\boldsymbol{A})\boldsymbol{x}\|_2^2\end{aligned}$$

仅当$\beta=\rho(\boldsymbol{x},\boldsymbol{A})$时上式为等式.

为确认$\boldsymbol{x}$和$\boldsymbol{A}\boldsymbol{x}-\rho(\boldsymbol{x},\boldsymbol{A})\boldsymbol{x}$的正交性，需验证

$$\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}-\rho(\boldsymbol{x},\boldsymbol{A})\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{x}-\frac{(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x})}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}\boldsymbol{x}\right)=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}=0.$$

正如所要求的那样. □

例5.5 利用例5.4的矩阵来说明定理5.5. 设$\boldsymbol{A}=\begin{bmatrix}1+g&\varepsilon\\\varepsilon&1\end{bmatrix}$，其中$0<\varepsilon<g$. 设$\boldsymbol{x}=[1,0]^{\mathrm{T}}$和$\beta=\rho(\boldsymbol{x},\boldsymbol{A})=1+g$. 则$r=\boldsymbol{A}\boldsymbol{x}-\beta\boldsymbol{x}=[0,\varepsilon]^{\mathrm{T}}$和$\|\boldsymbol{r}\|_2=\varepsilon$. $\boldsymbol{A}$的特征值是$\alpha_{\pm}=1+\frac{g}{2}\pm\frac{g}{2}\left(1+\left(\frac{2\varepsilon}{g}\right)^2\right)^{1/2}$，而特征向量在例5.4中给出(其中矩阵称为$\boldsymbol{A}+\boldsymbol{E}$而不是$\boldsymbol{A}$).

定理5.5预测$\|\boldsymbol{A}\boldsymbol{x}-\beta\boldsymbol{x}\|_2=\|\boldsymbol{r}\|_2=\varepsilon$是从$\beta=1+g$到$\boldsymbol{A}$的最近的特征值$\alpha_+$距离的一个界；这也可用外尔定理(定理5.1)预测. 下面将看到这个界比界(5.7)宽松得多.

当ε比g小得多时，将存在一个靠近$1+g$的特征值其特征向量靠近x，并且另一个特征值靠近1其特征向量靠近$[0,1]^{\mathrm{T}}$. 这意味着$\mathrm{gap}'=|\alpha_{-}-\rho(\boldsymbol{x},\boldsymbol{A})|=\frac{g}{2}\left(1+\left(1+\left(\frac{2\varepsilon}{g}\right)^2\right)^{1/2}\right)$，故界(5.6)推出$\boldsymbol{x}$和真正的特征向量之间夹角$\theta$以下式为界 [206]

$$\sin\theta \leqslant \frac{\|\boldsymbol{r}\|_2}{\text{gap}'} = \frac{2\varepsilon/g}{1+\left(1+\left(\frac{2\varepsilon}{g}\right)^2\right)^{1/2}}.$$

与例 5.4 中明确的特征向量比较，看出上界实际上等于 $\tan\theta$，对微小的 θ，它几乎与 $\sin\theta$ 相同. 故界(5.6)是十分精确的.

现对差 $|\beta-\alpha_+|$ 考察界(5.7). 对这个 2×2 阶矩阵例子证实 $|\beta-\alpha_+|$ 和它的界都精确地等于

$$\frac{\|\boldsymbol{r}\|_2^2}{\text{gap}'} = \varepsilon \cdot \frac{2\varepsilon/g}{1+\left(1+\left(\frac{2\varepsilon}{g}\right)^2\right)^{1/2}}.$$

在 $g=10^{-2}$ 和 $\varepsilon=10^{-5}$ 的特殊情况计算这些界. 于是 $\boldsymbol{A}$ 的特征值近似地为 $\alpha_+ = 1.010\,000\,01 = 1.01+10^{-8}$ 而 $\alpha_- = 0.999\,999\,99 = 1-10^{-8}$. 第一个界是 $|\beta-\alpha_+| \leqslant \|\boldsymbol{r}\|_2 = 10^{-5}$，它比实际的误差 10^{-8} 大 10^3 倍. 相反，界(5.7)是 $|\beta-\alpha_+| \leqslant \|\boldsymbol{r}\|_2^2/\text{gap}' = (10^{-5})^2/(1.01-\alpha_-) \approx 10^{-8}$，它是微小的. 对 α_+，x 和真正的特征向量之间的夹角大约是 10^{-3}，与界 $\|r\|_2/\text{gap}' = 10^{-5}/(1.01-\alpha_-) \approx 10^{-3}$ 一样. ◇

最后讨论当问题有一组 k 个紧密地成串的特征值并要计算它们的特征向量时会发生什么情况. “紧密地成串的”意指串中任意特征值与串中的特征值之间的间隙是微小的，但与不在串中的特征值是远离的. 例如，可能有 $k=20$ 个特征值在区间 $[0.9999, 1.0001]$ 中，而所有其他的特征值可能大于2. 因而定理5.4 和5.5 指出我们不能期待精确地得到各自的特征向量. 然而可能十分精确地计算由这些向量生成的 k-维不变子空间.

相对扰动理论

本节描述比上节中更紧凑的关于特征值和特征向量的界. 这些界对论证5.4.2 节和5.4.3 节中描述的计算奇异值和特征值的高精度算法是需要的.

为了对比将提出下面的界和前节中的那些界，考察 1×1 阶矩阵的情况. 给定一个标量 α，一个扰动的标量 $\hat{\alpha}=\alpha+e$ 和一个界 $|e|\leqslant\varepsilon$，可以明显地用 $|\hat{\alpha}-\alpha|\leqslant\varepsilon$. 来界定 $\hat{\alpha}$ 中的绝对误差. 这就是上节中采用的方法. 改为考察扰动的标量 $\hat{\alpha}=x^2\alpha$ 和一个界 $|x^2-1|\leqslant\varepsilon$. 用

$$\frac{|\hat{\alpha}-\alpha|}{|\alpha|} = |x^2-1| \leqslant \varepsilon.$$

[207] 来界定 $\hat{\alpha}$ 中的相对误差.

下面把这个简单的想法推广到矩阵. 在上节中我们用 $|\hat{\alpha}_i-\alpha_i| \leqslant \|\boldsymbol{E}\|_2$ 界定 $\boldsymbol{A}$ 的特征值 α_i 和 $\hat{\boldsymbol{A}}=\boldsymbol{A}+\boldsymbol{E}$ 的特征值 $\hat{\alpha}_i$ 中的绝对差异. 下面将根据 $\varepsilon \equiv \|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}-\boldsymbol{I}\|_2$ 来界定

A 的特征值 α_i 和 $\hat{A}=X^{\mathrm{T}}AX$ 的特征值 $\hat{\alpha}_i$ 之间的相对差异.

定理5.6 “相对”外尔定理. 设 A 有特征值 α_i 而 $\hat{A}=X^{\mathrm{T}}AX$ 有特征值 $\hat{\alpha}_i$. 设 $\varepsilon\equiv\|X^{\mathrm{T}}X-I\|_2$. 则 $|\hat{\alpha}_i-\alpha_i|\leqslant|\alpha_i|\varepsilon$. 若 $\alpha_i\neq0$, 则也可记作

$$\frac{|\hat{\alpha}_i-\alpha_i|}{|\alpha_i|}\leqslant\varepsilon. \tag{5.8}$$

证明 因为 $A-\alpha_iI$ 的第 i 个特征值为零. 所以西尔维斯特惯性定律告诉我们

$$X^{\mathrm{T}}(A-\alpha_iI)X=(X^{\mathrm{T}}AX-\alpha_iI)+\alpha_i(I-X^{\mathrm{T}}X)\equiv H+F.$$

的第 i 个特征值同样为零. 外尔定理断言 $|\lambda_i(H)-0|\leqslant\|F\|_2$ 或 $|\hat{\alpha}_i-\alpha_i|\leqslant|\alpha_i|\|X^{\mathrm{T}}X-I\|_2=|\alpha_i|\varepsilon$. □

注意, 当 X 正交时 $\varepsilon=\|X^{\mathrm{T}}X-I\|_2=0$, 故定理证实 $X^{\mathrm{T}}AX$ 和 A 有相同的特征值. 若 X “几乎”正交, 即 ε 是小的, 则定理断言在相对误差意义下特征值几乎相同.

推论5.2 设 G 是具有奇异值 σ_i 的任意矩阵, 并设 $\hat{G}=Y^{\mathrm{T}}GX$ 有奇异值 $\hat{\sigma}_i$. 设 $\varepsilon\equiv\max(\|X^{\mathrm{T}}X-I\|_2,\|Y^{\mathrm{T}}Y-I\|_2)$. 则 $|\hat{\sigma}_i-\sigma_i|\leqslant\varepsilon\sigma_i$. 若 $\sigma_i\neq0$, 则可记作

$$\frac{|\hat{\sigma}_i-\sigma_i|}{\sigma_i}\leqslant\varepsilon. \tag{5.9}$$

可类似地推广定理5.4去界定 A 的特征向量 q_i 和 $\hat{A}=X^{\mathrm{T}}AX$ 的特征向量 $\hat{q}_i$ 之间的差异. 为此需要定义谱中的相对间隙(relative gap).

定义5.4 A 的一个特征值 α_i 与谱的其余特征值之间的相对间隙定义为 $\mathrm{rel_gap}(i,A)=\min_{j\neq i}\frac{|\alpha_j-\alpha_i|}{|\alpha_i|}$.

定理5.7 假如 A 有特征值 α_i 和相应的单位特征向量 q_i. 假如 $\hat{A}=X^{\mathrm{T}}AX$ 有特征值 $\hat{\alpha}_i$ 和相应的单位特征向量 $\hat{q}_i$. 设 θ 是 q_i 和 $\hat{q}_i$ 之间的锐角. 设 $\varepsilon_1=\|I-X^{-\mathrm{T}}X^{-1}\|_2,\varepsilon_2=\|X-I\|_2$. 倘若 $\varepsilon_1<1$ 且 $\mathrm{rel_gap}(i,X^{\mathrm{T}}AX)>0$, 则

$$\frac{1}{2}\sin2\theta\leqslant\frac{\varepsilon_1}{1-\varepsilon_1}\cdot\frac{1}{\mathrm{rel_gap}(i,\ X^{\mathrm{T}}AX)}+\varepsilon_2.$$

208

证明 设 $\eta=\hat{\alpha}_i-\alpha_i,H=A-\hat{\alpha}_iI,F=\hat{\alpha}_i(I-X^{-\mathrm{T}}X^{-1})$. 注意

$$H+F=A-\hat{\alpha}_iX^{-\mathrm{T}}X^{-1}=X^{-\mathrm{T}}(X^{\mathrm{T}}AX-\hat{\alpha}_iI)X^{-1}.$$

因而 $Hq_i=-\eta q_i$ 和 $(H+F)(X\hat{q}_i)=0$, 所以 $X\hat{q}_i$ 是 $H+F$ 第 i 个特征值0的一个特征向量. 设 θ_1 是 q_i 和 $X\hat{q}_i$ 之间的锐角. 由定理5.4, 可以界定

$$\frac{1}{2}\sin2\theta_1\leqslant\frac{\|F\|_2}{\mathrm{gap}(i,H+F)}. \tag{5.10}$$

我们有 $\|F\|_2=|\hat{\alpha}_i|\varepsilon_1$. 现在 $\mathrm{gap}(i,H+F)$ 是 $H+F$ 最小非零特征值的大小. 因为 $X^{\mathrm{T}}(H+F)X=X^{\mathrm{T}}AX-\hat{\alpha}_iI$ 有特征值 $\hat{\alpha}_j-\hat{\alpha}_i$, 所以定理5.6告诉我们 $H+F$ 的特征值位于

从$(1-\varepsilon_1)(\hat{\alpha}_j-\hat{\alpha}_c)$到$(1+\varepsilon_1)(\hat{\alpha}_j-\hat{\alpha}_i)$的区间内. 因而$\mathrm{gap}(i,\boldsymbol{H}+\boldsymbol{F})\geqslant(1-\varepsilon_1)\mathrm{gap}(i,\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X})$, 故代入到(5.10)中得到

$$\frac{1}{2}\sin 2\theta_1\leqslant\frac{\varepsilon_1|\hat{\alpha}_i|}{(1-\varepsilon_1)\mathrm{gap}(i,\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X})}=\frac{\varepsilon_1}{(1-\varepsilon_1)\mathrm{rel_gap}\ (i,\ \boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X})}.\tag{5.11}$$

现在设θ_2是$\boldsymbol{X}\hat{\boldsymbol{q}}_i$和$\hat{\boldsymbol{q}}_i$之间的锐角, 所以$\theta\leqslant\theta_1+\theta_2$. 利用三角学可以界定$\sin\theta_2\leqslant\|(\boldsymbol{X}-\boldsymbol{I})\hat{\boldsymbol{q}}_i\|_2\leqslant\|\boldsymbol{X}-\boldsymbol{I}\|_2=\varepsilon_2$, 故利用三角不等式(见问题5.11)可得

$$\begin{aligned}\frac{1}{2}\sin 2\theta&\leqslant\frac{1}{2}\sin 2\theta_1+\frac{1}{2}\sin 2\theta_2\\&\leqslant\frac{1}{2}\sin 2\theta_1+\sin\theta_2\\&\leqslant\frac{\varepsilon_1}{(1-\varepsilon_1)\mathrm{rel_gap}\ (i,\ \boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X})}+\varepsilon_2\end{aligned}$$

证毕. □

对奇异向量可以证明一个类似的定理[101].

例 5.6　再次考察例 5.1 的质点-弹簧系统并利用它说明外尔定理(定理 5.1)提供的特征值的界比外尔定理的"相对"形式(定理 5.6)提供的界差(宽松)得多. 还将看到定理 5.7 的特征向量的界比定理 5.4 的界好(紧凑)得多.

假如$\boldsymbol{M}=\mathrm{diag}(1,100,10\,000)$、$\boldsymbol{K}_D=\mathrm{diag}(10\,000,100,1)$. 仿效例 5.1, 定义$\boldsymbol{K}=\boldsymbol{B}\boldsymbol{K}_D\boldsymbol{B}^{\mathrm{T}}$, $\hat{\boldsymbol{K}}=\boldsymbol{M}^{-1/2}\boldsymbol{K}\boldsymbol{M}^{-1/2}$, 其中

$$\boldsymbol{B}=\begin{bmatrix}1&-1&&\\&\ddots&\ddots&\\&&\ddots&-1\\&&&1\end{bmatrix}$$

209

故

$$\hat{\boldsymbol{K}}=\boldsymbol{M}^{-1/2}\boldsymbol{K}\boldsymbol{M}^{-1/2}=\begin{bmatrix}10\,100&-10&\\-10&1.01&-0.001\\&-0.001&0.0001\end{bmatrix}.$$

至 5 位小数, $\hat{\boldsymbol{K}}$的特征值是 10 100, 1.0001 和 0.000 99 假定质量(m_{ii})和弹簧常数($k_{D,ii}$)每个至多扰动 1%. 试问特征值可能改变多少? 最大的矩阵元素是$\hat{\boldsymbol{K}}_{11}$, m_{11}改变为 0.99, $k_{D,11}$改变为 10 100, $\hat{K}_{11}$将改变为 10 305 左右, 平均改变 205. 于是, 外尔定理告诉我们每个特征值可能改变差不多 ±205, 它将彻底地改变两个较小的特征值. 由定理 5.4 得到的特征向量的界也指出相应的特征向量会完全地改变.

现在应用定理 5.6 于$\hat{\boldsymbol{K}}$, 或实际上应用推论 5.2 于$\boldsymbol{G}=\boldsymbol{M}^{-1/2}\boldsymbol{B}\boldsymbol{K}_D^{1/2}$, 其中$\hat{\boldsymbol{K}}=\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}}$如例 5.1 中那样定义. 改变每个质量至多 1% 等价于扰动$\boldsymbol{G}$为$\boldsymbol{X}\boldsymbol{G}$, 其中$\boldsymbol{X}$是对角元

在 $1/\sqrt{0.99}\approx1.005$ 至 $1/\sqrt{1.01}\approx0.995$ 之间的对角阵. 于是推论 5.2 告诉我们 $\boldsymbol{G}$ 的奇异值只会改变区间[0.995,1.005]中的一个因子，故 $\boldsymbol{M}$ 的特征值也只会改变 1%. 换言之，最小的特征值正如最大的特征值一样只会改变它的第 2 位小数. 类似地，改变弹簧常数至多 1% 等价于改变 $\boldsymbol{G}$ 为 $\boldsymbol{GX}$，特征值的改变还是不会超过 1%. 若同时扰动 $\boldsymbol{M}$ 和 $\boldsymbol{K}_D$，则特征值将移动 2% 左右. 因为特征值大小差别如此之大，它们的相对间隙都十分大，所以它们的特征向量也只可能旋转角度 3% 左右. ◇

更适合于由微分("无界")算子产生的矩阵的一个不同的相对误差分析方法可见 [161].

5.3 对称特征问题的算法

我们讨论对称特征问题的多种算法. 如概述中提及的那样，这里只讨论*直接法*，*迭代法*留在第 7 章中讨论.

在第 4 章中对非对称特征问题我们仅讨论的 QR 算法可以求所有的特征值和所有可选的特征向量. 关于对称特征问题已经有很多可以利用的算法，这些算法提供给我们更多的灵活性和有效性. 例如，下面描述的*对分法*可用于只求用户指定的区间 $[a,b]$ 中的特征值，这个方法可以做得比它求所有的特征值快得多.

除了瑞利商迭代和雅可比方法外，下面所有的算法都假定利用 4.4.7 节中算法 4.6 的变形首先把矩阵化为三对角形式. 这个初始代价为 $\frac{4}{3}n^3$ flops，若特征向量也要求的话代价为 $\frac{8}{3}n^3$ flops. 210

1. *三对角 QR 迭代*(tridiagonal QR iteration) 这个算法求对称三对角阵的所有特征值和所有可选的特征向量. 它目前是求对称三对角阵所有特征值最快的实用方法，高效地执行 $O(n^2)$ flops. 因为约化一个稠密阵为三对角阵的代价为 $\frac{4}{3}n^3$ flops，所以对足够大的 n, $O(n^2)$ 是可以忽略不计的，但是还要求所有特征向量的话，QR 迭代平均稍稍多于 $6n^3$ flops，所以它仅对直到 $n=25$ 左右为止的小矩阵是最快的算法. 这是构成 Matlab 指令 `eig`[1] 和 LAPACK 程序 `ssyev`(关于稠密阵)和 `sstev`(关于三对角阵)基础的算法.

2. *瑞利商迭代*(Rayleigh quotient iteration) 这个算法成为 QR 迭代的基础，但是为了更容易地分析其特别迅速的收敛性并且它本身也可作为一个算法，所以独立地提出这个算法. 事实上，它一般是立方收敛(如同做 QR 迭代)，这意味着在每一步正确的数字位数渐近地增至三倍.

3. *分而治之*(Divide-and-conquer) 这个算法目前是求大于 $n=25$ 的对称三对角阵所有特征值和特征向量的最快速方法. (在 LAPACK 中执行 `sstevd`，对较小的矩

1. Matlab 检查 `eig` 的自变量是否对称，当满足条件时使用对称算法.

阵默认 QR 迭代).

在最坏的情况下，分而治之需要 $O(n^3)$ flops，但是实际上常数非常小. 经过一大组随机测试情况，发现它平均只取 $O(n^{2.3})$ flops，对某些特征值分布如 $O(n^2)$ 那样低.

理论上，可运行 $O(n \cdot \log^p n)$ flops 来执行分而治之，其中 p 是一个小的整数 [131]. 这个超快的实现使用快速多极方法(FMM)[124]，后者最初是用于计算 n 个电负荷粒子相互作用力的完全不同的问题. 但是这个超快的实现的复杂性意味着 QR 迭代目前是求所有特征值首选的算法，而没有 FMM 的分而治之是求所有特征值和所有特征向量首选的方法.

4. *对分法和逆迭代*(Bisection and inverse iteration)　对分法可用于只求对称三对角阵特征值的一个子集，譬如求那些在区间 $[a,b]$ 或 $[\alpha_i,\alpha_{i-j}]$ 中的特征值. 它只需要 $O(nk)$ flops，其中 k 是所要求的特征值个数. 因而当 $k \ll n$ 时，对分法可比 QR 迭代快得多，因为 QR 迭代需要 $O(n^2)$ flops. 逆迭代(算法 4.2)可用于求相应的特征向量. 在最佳的情况，当特征值“适当地分离”时(后面将更完整地说明这点)，逆迭代也仅花费 $O(nk)$ flops. 甚至要求所有特征值和特征向量时($k=n$)，这个算法不是比 QR 就是比分而治之(没有 FMM)花费少很多. 但是在最坏的情况，当许多特征值是成串地紧靠在一起时，逆迭代花费 $O(nk^2)$ flops，而且它甚至不保证计算的特征向量的精度(虽然实际上它几乎总是精确的). 故分而治之和 QR 目前是求所有的(或大部分的)特征值和特征向量特别当特征值可能成串时首选的方法. 对分法和逆迭代作为 LAPACK 程序 ssyevx 中的选项是可用的.

211

正在进行处理稠密的特征值问题的逆迭代法的最新研究，有可能使逆迭代法成为求所有特征向量的最迅速的方法(理论上还有 FMM 的分而治之方法)[105,203,83,201,176,173,269]. 然而，还没有可用的实现这个逆迭代改进形式的软件.

5. *雅可比方法*　这个方法是特征问题最古老的方法，始于 1846 年. 它通常比上面任何方法都慢，它需要 $O(n^3)$ flops 且带一个大的常数. 但是这个方法仍具有吸引力，因为它有时比上面任何方法都要精确得多. 这是因为雅可比方法有时有可能达到 5.2.1 节中描述的相对精度，并且有时能比前面的方法更精确地计算微小的特征值[82]. 在 5.4.3 节中讨论雅可比方法的高精度性质，那里我们指出如何计算 SVD.

随后几节更详尽地描述这些算法. 5.3.6 节提供了性能的比较.

5.3.1　三对角 QR 迭代

回顾非对称特征问题 QR 算法有两个阶段:

1. 给定 $\boldsymbol{A}$，利用算法 4.6 求一个正交阵 $\boldsymbol{Q}$ 使得 $\boldsymbol{QAQ}^T=\boldsymbol{H}$ 是上海森伯格阵.

2. 对 $\boldsymbol{H}$ 应用 QR 迭代(如 4.4.8 节所述)，得到收敛于实舒尔型的上海森伯格阵序列 $\boldsymbol{H}=\boldsymbol{H}_0,\boldsymbol{H}_1,\boldsymbol{H}_2,\cdots$.

212

关于对称特征问题的第一个算法完全类似于这个算法：

1. 给定 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$，利用4.4.7节中的算法4.6的变形求一个正交阵 $\boldsymbol{Q}$ 使得 $\boldsymbol{QAQ}^{\mathrm{T}}=\boldsymbol{T}$ 是三对角阵.

2. 对 $\boldsymbol{T}$ 应用QR迭代得到收敛于对角型的三对角阵序列 $\boldsymbol{T}=\boldsymbol{T}_0,\boldsymbol{T}_1,\boldsymbol{T}_2,\cdots$.

注意因为 $\boldsymbol{QAQ}^{\mathrm{T}}$ 是对称和上海森伯格的，所以它必定也是下海森伯格的，即三对角的，我们可以看出QR迭代保持所有的 $\boldsymbol{T}_i$ 三对角. 这就使每次QR迭代非常便宜. 运算量呈现如下：

1. 化 $\boldsymbol{A}$ 为对称三对角形式 $\boldsymbol{T}$ 花费 $\frac{4}{3}n^3+O(n^2)$ flops，若再求特征向量将花费 $\frac{8}{3}n^3+O(n^2)$ flops.

2. 带单个位移的一次三对角QR迭代（“凸出部分驱赶”）花费 $6n$flops.

3. 求 $\boldsymbol{T}$ 的所有特征值每个特征值平均只做2次QR步，总计 $6n^2$flops.

4. 求 $\boldsymbol{T}$ 的所有特征值和特征向量花费 $6n^3+O(n^2)$ flops.

5. 仅求 $\boldsymbol{A}$ 的特征值将花费 $\frac{4}{3}n^3+O(n^2)$ flops.

6. 求 $\boldsymbol{A}$ 的所有特征值和特征向量总共花费 $8\frac{2}{3}n^3+O(n^2)$ flops.

还需描述如何选取位移来实施每次QR迭代. 用

$$\boldsymbol{T}_i=\begin{bmatrix} a_1 & b_1 & & \\ b_1 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n \end{bmatrix}$$

表示第 i 次迭代. 位移的最简单选择应该是 $\sigma_i=a_n$；这是4.4.8节中讨论的单个位移QR迭代. 结果如下节中指出的那样几乎对所有的矩阵是立方收敛的. 遗憾地，存在其不收敛的例子[197,p.76]，所以为得到整体收敛性需要稍微更复杂一点的位移策略：设位移 σ_i 是最接近于 a_n 的 $\begin{bmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{bmatrix}$ 的特征值. 这个位移称为威尔金森位移(Wilkinson's shift).

定理5.8 威尔金森(Wilkinson). *具有威尔金森位移的QR迭代是整体收敛的并且至少是线性收敛的. 对几乎所有的矩阵它是渐近立方收敛的.*

这个定理的证明可在[197]中找到. 这个程序可利用LAPACK中的 `ssyev`. 当只要求特征值时(`ssterf`；见[104,200])比还要计算特征向量时(`ssteqr`)算法的内循环可以组织得更为有效.

例5.7 下面是三对角QR迭代收敛性的一个说明，用下列三对角阵开始（只指出对角线，按列表示）：

$$
T_0 = \mathrm{tridiag}\begin{bmatrix} & 0.24929 & \\ 1.263 & & 1.263 \\ & 0.96880 & \\ -0.82812 & & -0.82812 \\ & 0.48539 & \\ -3.1883 & & -3.1883 \\ & -0.91563 & \end{bmatrix}.
$$

下列表格指出每个 T_i 最后的非对角元，每个 T_i 最后的对角元，以及最后的对角元与它的最终值(特征值 $\alpha\approx-3.54627$)之间的差. 在最后一行中误差立方收敛于零是明显的.

i	$T_i(4,3)$	$T_i(4,4)$	$T_i(4,4)-\alpha$
0	-3.1883	-0.91563	2.6306
1	$-5.7\cdot10^{-2}$	-3.5457	$5.4\cdot10^{-4}$
2	$-2.5\cdot10^{-7}$	-3.5463	$1.2\cdot10^{-14}$
3	$-6.1\cdot10^{-23}$	-3.5463	0

在这一点上

$$
T_3 = \mathrm{tridiag}\begin{bmatrix} & 1.9871 & \\ 0.77513 & & 0.77513 \\ & 1.7049 & \\ -1.7207 & & -1.7207 \\ & 0.64214 & \\ -6.1\cdot10^{-23} & & -6.1\cdot10^{-23} \\ & -3.5463 & \end{bmatrix},
$$

我们置(4.3)和(3.4)上的非常微小的元素为0. 这称为收缩(deflation)，仅仅按 $6.1\cdot10^{-23}$ 标准扰动 T_3 这个算法是稳定的. 现在再对 T_3 的前 3×3 阶子阵应用QR迭代，重复此过程得到其他的特征值. 见 HOMEPAGE/Matlab/tridiQR.m.

5.3.2 瑞利商迭代

回顾4.4节中我们不明显地在每步做逆迭代的QR迭代分析. 当用于逆迭代中选择的位移是瑞利商时，我们更细致地加以研究.

214

算法5.1 瑞利商迭代：给定 x_0, $\|x_0\|_2=1$ 以及一个用户提供的停止容限tol，作迭代

$\rho_0=\rho(x_0,A)=\dfrac{x_0^{\mathrm T}Ax_0}{x_0^{\mathrm T}x_0}$

$i=0$

repeat

　$y_i=(A-\rho_{i-1}I)^{-1}x_{i-1}$

（续）

$x_i = y_i / \|y_i\|_2$ $\rho_i = \rho(x_i, A)$ $i = i+1$ until convergence($\|Ax_i - \rho_i x_i\|_2 < \text{tol}$)

当满足停止准则时，定理5.5告诉我们ρ_i是不超出tol的A的一个特征值.

若在QR迭代中使用位移$\sigma_i = a_{nn}$并用$x_0 = [0,\cdots,0,1]^T$起始作瑞利商迭代，则4.4节中讨论的QR和逆迭代之间联系可用于证明来自两个算法的σ_i和ρ_i序列是完全相同的(见问题5.13). 此时，我们将证明几乎总是立方收敛的.

定理5.9 瑞利商迭代是局部立方收敛的，即一旦误差足够小并且特征值是单重的，每步上正确数字的个数增至三倍.

证明 我们断言分析A为对角阵的情况就足够了. 为观察原因，设$Q^TAQ = \Lambda$，其中Q的列是特征向量的正交阵，而$\Lambda = \text{diag}(\alpha_1,\cdots,\alpha_n)$是特征值的对角阵. 现改变瑞利商迭代中的变量为$\hat{x}_i \equiv Q^T x_i$和$\hat{y}_i \equiv Q^T y_i$. 则

$$\rho_i = \rho(x_i, A) = \frac{x_i^T A x_i}{x_i^T x_i} = \frac{\hat{x}_i^T Q^T A Q \hat{x}_i}{\hat{x}_i^T Q^T Q \hat{x}_i} = \frac{\hat{x}_i^T \Lambda \hat{x}_i}{\hat{x}_i^T \hat{x}_i} = \rho(\hat{x}_i, \Lambda)$$

且$Q\hat{y}_{i+1} = (A - \rho_i I)^{-1} Q\hat{x}_i$，故

$$\hat{y}_{i+1} = Q^T (A - \rho_i I)^{-1} Q\hat{x}_i = (Q^T A Q - \rho_i I)^{-1} \hat{x}_i = (\Lambda - \rho_i I)^{-1} \hat{x}_i.$$

所以，对A和x_0运行瑞利商迭代等价于对Λ和$\hat{x}_0$运行瑞利商迭代. 因此不失一般性，我们假定$A = \Lambda$已经为对角阵. 故A的特征向量为单位阵的列e_i.

不失一般性假定x_i收敛于e_1，所以可记$x_i = e_1 + d_i$，其中$\|d_i\|_2 \equiv \varepsilon \ll 1$. 为证明立方收敛，我们需证明$x_{i+1} = e_1 + d_{i+1}$，$\|d_{i+1}\|_2 = O(\varepsilon^3)$.

首先注意

$$1 = x_i^T x_i = (e_1 + d_i)^T (e_1 + d_i) = e_1^T e_1 + 2e_1^T d_i + d_i^T d_i = 1 + 2d_{i1} + \varepsilon^2$$

215

以致$d_{i1} = -\varepsilon^2/2$. 所以

$$\rho_i = x_i^T \Lambda x_i = (e_1 + d_i)^T \Lambda (e_1 + d_i) = e_1^T \Lambda e_1 + 2e_1^T \Lambda d_i + d_i^T \Lambda d_i = \alpha_1 - \eta,$$

其中$\eta \equiv -2e_1^T \Lambda d_i - d_i^T \Lambda d_i = \alpha_1 \varepsilon^2 - d_i^T \Lambda d_i$. 我们看到

$$|\eta| \leqslant |\alpha_1| \varepsilon^2 + \|\Lambda\|_2 \|d_i\|_2^2 \leqslant 2\|\Lambda\|_2 \varepsilon^2, \tag{5.12}$$

故$\rho_i = \alpha_1 - \eta = \alpha_1 + O(\varepsilon^2)$是特征值$\alpha_1$的一个非常好的近似.

现在可记

$$\begin{aligned} y_{i+1} &= (\Lambda - \rho_i I)^{-1} x_i \\ &= \left[\frac{x_{i1}}{\alpha_1 - \rho_i}, \frac{x_{i2}}{\alpha_2 - \rho_i}, \cdots, \frac{x_{in}}{\alpha_n - \rho_i}\right]^T \qquad \text{因为} (\Lambda - \rho_i I)^{-1} = \text{diag}\left(\frac{1}{\alpha_j - \rho_i}\right) \\ &= \left[\frac{1 + d_{i1}}{\alpha_1 - \rho_i}, \frac{d_{i2}}{\alpha_2 - \rho_i}, \cdots, \frac{d_{in}}{\alpha_n - \rho_i}\right]^T \qquad \text{因为} x_i = e_1 + d_i \end{aligned}$$

$$= \left[\frac{1-\varepsilon^2/2}{\eta}, \frac{d_{i2}}{\alpha_2 - \alpha_1 + \eta}, \cdots, \frac{d_{in}}{\alpha_n - \alpha_1 + \eta} \right]^{\mathrm{T}} \quad \text{因为 } \rho_i = \alpha_1 - \eta \text{ 和 } d_{i1} = -\varepsilon^2/2$$

$$= \frac{1-\varepsilon^2/2}{\eta} \cdot \left[1, \frac{d_{i2}\eta}{(1-\varepsilon^2/2)(\alpha_2 - \alpha_1 + \eta)}, \cdots, \frac{d_{in}\eta}{(1-\varepsilon^2/2)(\alpha_n - \alpha_1 + \eta)} \right]^{\mathrm{T}}$$

$$\equiv \frac{1-\varepsilon^2/2}{\eta} \cdot (\boldsymbol{e}_1 + \hat{\boldsymbol{d}}_{i+1}).$$

为了界定$\|\hat{\boldsymbol{d}}_{i+1}\|_2$，注意利用$|\alpha_j - \alpha_1 + \eta| \geqslant \mathrm{gap}(1, \boldsymbol{\Lambda}) - |\eta|$可界定每个分母，另外又利用(5.12)得到

$$\|\hat{\boldsymbol{d}}_{i+1}\|_2 \leqslant \frac{\|\boldsymbol{d}_i\|_2 |\eta|}{(1-\varepsilon^2/2)(\mathrm{gap}(1,\boldsymbol{\Lambda}) - |\eta|)} \leqslant \frac{2\|\boldsymbol{\Lambda}\|_2 \varepsilon^3}{(1-\varepsilon^2/2)(\mathrm{gap}(1,\boldsymbol{\Lambda}) - 2\|\boldsymbol{\Lambda}\|\varepsilon^2)}$$

或$\|\hat{\boldsymbol{d}}_{i+1}\|_2 = O(\varepsilon^3)$. 最后，因为$\boldsymbol{x}_{i+1} = \boldsymbol{e}_1 + \boldsymbol{d}_{i+1} = (\boldsymbol{e}_1 + \hat{\boldsymbol{d}}_{i+1}) / \|\boldsymbol{e}_1 + \hat{\boldsymbol{d}}_{i+1}\|_2$，所以又看到$\|d_{i+1}\|_2 = O(\varepsilon^3)$. □

5.3.3　分而治之

如果你要计算维数大于 25 左右的三对角阵的所有特征值和特征向量，分而治之是现今可利用的最快速的方法(精确的阈值依赖于计算机). 它以一种数值稳定的方式执行是十分巧妙的. 虽然这个方法在 1981 年首次引进[59]，实际上直到 1992 年还未发现“正确的”执行过程[127,131]. 对稠密矩阵这个程序可利用 LAPACK 程序 ssyevd，对三对角阵可利用 sstevd. 这个程序对维数大于 25 的矩阵使用分而治之而对较小的矩阵(或者只要求特征值)自动地转换到 QR 迭代.

216

首先讨论算法的总体结构，数值上的细节留在后面讨论. 设

$$\boldsymbol{T} = \left[\begin{array}{cccc|cccc} a_1 & b_1 & & & & & & \\ b_1 & \ddots & \ddots & & & & & \\ & \ddots & a_{m-1} & b_{m-1} & & & & \\ & & b_{m-1} & a_m & b_m & & & \\ \hline & & & b_m & a_{m+1} & b_{m+1} & & \\ & & & & b_{m+1} & \ddots & \ddots & b_{n-1} \\ & & & & & & b_{n-1} & a_n \end{array}\right]$$

$$= \left[\begin{array}{cccc|cccc} a_1 & b_1 & & & & & & \\ b_1 & \ddots & \ddots & & & & & \\ & \ddots & a_{m-1} & b_{m-1} & & & & \\ & & b_{m-1} & a_m - b_m & & & & \\ \hline & & & & a_{m+1} - b_m & b_{m+1} & & \\ & & & & b_{m+1} & \ddots & \ddots & b_{n-1} \\ & & & & & & b_{n-1} & a_n \end{array}\right]$$

$$+\left[\begin{array}{c|c} b_m & b_m \\ \hline b_m & b_m \end{array}\right]$$

$$=\left[\begin{array}{c|c} T_1 & 0 \\ \hline 0 & T_2 \end{array}\right]+b_m\cdot\begin{bmatrix}0\\ \vdots\\ 0\\ 1\\ 1\\ 0\\ \vdots\\ 0\end{bmatrix}[0,\cdots,0,1,1,0,\cdots,0]\equiv\left[\begin{array}{c|c} T_1 & 0 \\ \hline 0 & T_2 \end{array}\right]+b_m vv^{\mathrm{T}}.$$

217

假定已经有 T_1 和 T_2 的特征分解：$T_i=Q_i\Lambda_iQ_i^{\mathrm{T}}$. 这些将通过这个相同的算法递归地计算. 我们叙述 T 的特征值与 T_1 和 T_2 的特征值的关系如下：

$$\begin{aligned}T&=\begin{bmatrix}T_1 & 0\\ 0 & T_2\end{bmatrix}+b_m vv^{\mathrm{T}}\\ &=\begin{bmatrix}Q_1\Lambda_1Q_1^{\mathrm{T}} & 0\\ 0 & Q_2\Lambda_2Q_2^{\mathrm{T}}\end{bmatrix}+b_m vv^{\mathrm{T}}\\ &=\begin{bmatrix}Q_1 & 0\\ 0 & Q_2\end{bmatrix}\left(\begin{bmatrix}\Lambda_1 & \\ & \Lambda_2\end{bmatrix}+b_m uu^{\mathrm{T}}\right)\begin{bmatrix}Q_1^{\mathrm{T}} & 0\\ 0 & Q_2^{\mathrm{T}}\end{bmatrix},\end{aligned}$$

其中

$$u=\begin{bmatrix}Q_1^{\mathrm{T}} & 0\\ 0 & Q_2^{\mathrm{T}}\end{bmatrix}\quad v=\begin{bmatrix}Q_1^{\mathrm{T}}\text{ 的最后列}\\ Q_2^{\mathrm{T}}\text{ 的第一列}\end{bmatrix}$$

因为 $v=[0,\cdots,0,1,1,0,\cdots,0]^{\mathrm{T}}$. 因此，$T$ 的特征值和相似矩阵 $D+\rho uu^{\mathrm{T}}$ 的那些特征值相同，其中 $D=\begin{bmatrix}\Lambda_1 & 0\\ 0 & \Lambda_2\end{bmatrix}$是对角阵，$\rho=b_m$ 是标量而 u 是向量. 不失一般性，今后将假定 D 的对角元 $d_1,\cdots,d_n$ 整理成：$d_n\leqslant\cdots\leqslant d_1$.

为求 $D+\rho uu^{\mathrm{T}}$ 的特征值，首先假定 $D-\lambda I$ 非奇异，并计算特征多项式如下：

$$\det(D+\rho uu^{\mathrm{T}}-\lambda I)=\det((D-\lambda I)(I+\rho(D-\lambda I)^{-1}uu^{\mathrm{T}})).\tag{5.13}$$

因为 $D-\lambda I$ 非奇异，所以每当 λ 是一个特征值时 $\det(I+\rho(D-\lambda I)^{-1}uu^{\mathrm{T}})=0$. 注意 $I+\rho(D-\lambda I)^{-1}uu^{\mathrm{T}}$ 是一个单位阵加一个秩-1 矩阵；这种矩阵的行列式是容易计算的.

引理 5.1 *若 x 和 y 是向量，则* $\det(I+xy^{\mathrm{T}})=1+y^{\mathrm{T}}x$.

证明留至问题 5.14.

所以

$$\det(I+\rho(D-\lambda I)^{-1}uu^{\mathrm{T}}) = 1+\rho u^{\mathrm{T}}(D-\lambda I)^{-1}u$$
$$= 1+\rho\sum_{i=1}^{n}\frac{u_i^2}{d_i-\lambda} \equiv f(\lambda), \tag{5.14}$$

T 的特征值是所谓特征方程(secular equation)$f(\lambda)=0$ 的根. 若所有的 d_i 是相异的而且所有的 $u_i\neq 0$(一般的情况), 则函数 $f(\lambda)$ 具有图5-2所示的图像($n=4$ 和 $\rho>0$).

正如我们所见, 线 $y=1$ 是一条水平渐近线, 而线 $\lambda=d_i$ 是垂直渐近线. 因为 $f'(\lambda)=\rho\sum_{i=1}^{n}\frac{u_i^2}{(d_i-\lambda)^2}>0$, 所以除了在线 $\lambda=d_i$ 之外函数是严格递增的. 因此, $f(\lambda)$ 的根由 d_i 错开, 并且在 d_1 的右边有额外的一个根(图5-2中 $d_1=4$). (若 $\rho<0$, 则 $f(\lambda)$ 递减, 在 d_n 的左边有额外的一个根). 因为 $f(\lambda)$ 在区间 (d_i,d_{i+1}) 上单调和光滑的, 所以给定 (d_i,d_{i+1}) 中的一个初始点可能找到一个快速和单调收敛于每个根的牛顿法的变形. 我们在本节后面部分讨论其细节. 这里我们必须知道的是实际上对每个特征值牛顿法以一个有界的步数收敛. 因为 $f(\lambda)$ 和 $f'(\lambda)$ 求值花费 $O(n)$ flops, 求一个特征值花费 $O(n)$ flops, 故求 $D+\rho uu^{\mathrm{T}}$ 的全部 n 个特征值花费 $O(n^2)$ flops.

[218]

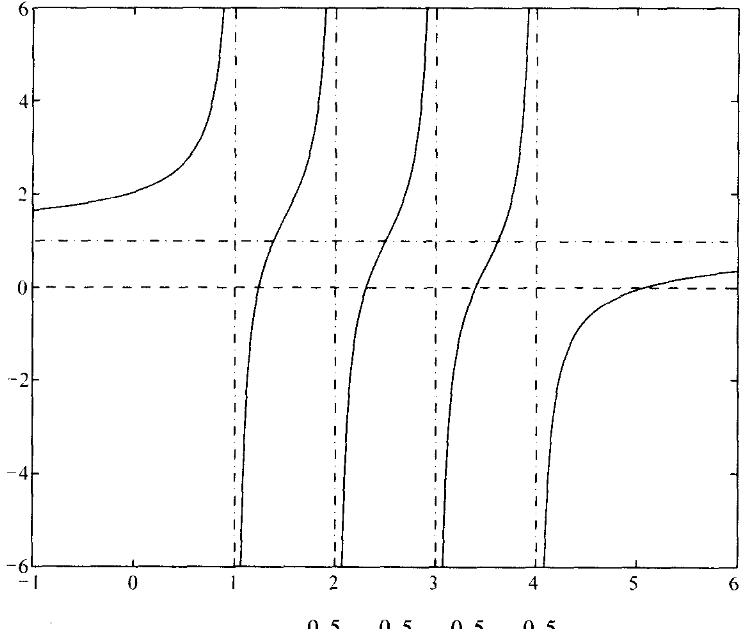


图5-2　$f(\lambda)=1+\frac{0.5}{1-\lambda}+\frac{0.5}{2-\lambda}+\frac{0.5}{3-\lambda}+\frac{0.5}{4-\lambda}$ 的图像

导出 $D+uu^{\mathrm{T}}$ 的特征向量的表达式也是容易的.

引理5.2　若 α 是 $D+\rho uu^{\mathrm{T}}$ 的一个特征值, 则 $(D-\alpha I)^{-1}u$ 是其特征向量. 因为 $D-$

αI 是对角阵，所以计算花费 $O(n)$ flops.

证明

$$(D+\rho uu^T)[(D-\alpha I)^{-1}u] = (D-\alpha I+\alpha I+\rho uu^T)(D-\alpha I)^{-1}u$$
$$= u+\alpha(D-\alpha I)^{-1}u+u[\rho u^T(D-\alpha I)^{-1}u]$$
$$= u+\alpha(D-\alpha I)^{-1}u-u$$
$$\text{因为 } \rho u^T(D-\alpha I)^{-1}u+1=f(\alpha)=0$$
$$= \alpha[(D-\alpha I)^{-1}u].$$

证毕. □

这个公式计算所有 n 个特征向量花费 $O(n^2)$ flops. 遗憾地，这个简单的特征向量公式不是数值稳定的，因为两个非常接近的 α_i 值可能导致非正交的计算的特征向量 219 u_i. 从这个算法的原来公式寻找一个稳定的替代公式用了十多年时间. 我们在本节后面部分讨论其细节.

全部的算法是递归的.

算法 5.2 利用分而治之求对称三对角阵之特征值和特征向量.

proc dc_eig(T,Q,Λ)⋯. 从输入 T 计算输出 Q 和 Λ，其中 $T=Q\Lambda Q^T$

if T is 1×1

  return $Q=1,\Lambda=T$

else

  形成 $T=\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}+b_m vv^T$

  call dc_eig(T_1, Q_1, Λ_1)

  call dc_eig(T_2, Q_2, Λ_2)

  从 Λ_1, Λ_2, Q_1, Q_2 形成 $D+\rho uu^T$

  求 $D+\rho uu^T$ 的特征值 Λ 和特征向量 Q'

  形成 $Q=\begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}\cdot Q'=T$ 的特征向量

  return Q 和 Λ

endif

分析算法 5.2 的复杂性如下. 设 $t(n)$ 是对 $n\times n$ 阶矩阵运行 *dc_eig* 的 flops 数. 则

$t(n)=2t(n/2)$ 2 次递归调用 $dc_eig(T_i,Q_i,\Lambda_i)$

$+O(n^2)$ 求 $D+\rho uu^T$ 的特征值

$+O(n^2)$ 求 $D+\rho uu^T$ 的特征向量

$$+c\cdot n^3 \qquad \text{矩阵乘法 } \boldsymbol{Q}=\begin{bmatrix}\boldsymbol{Q}_1 & 0\\ 0 & \boldsymbol{Q}_2\end{bmatrix}\cdot \boldsymbol{Q}'.$$

若把 $\boldsymbol{Q}_1$，$\boldsymbol{Q}_2$ 和 $\boldsymbol{Q}'$ 当作稠密阵并利用标准的矩阵乘法算法，则最后一行中的常数 $c=1$. 因而看出算法中的主要代价是最后一行中的矩阵乘法. 不计 $O(n^2)$ 项的话，得到 $t(n)=2t(n/2)+cn^3$. 可以计算这个几何和，得到 $t(n)\approx c\frac{4}{3}n^3$（见问题5.15）. 实际上，$c$ 通常比1小得多，因为一个称为收缩(deflation)的现象使 Q' 非常稀疏.

在下节中讨论收缩之后，讨论特征方程的求解并稳定地计算特征向量. 最后讨论如何用静电粒子模拟中开发的FMM技术去加速方法[124]. 这几节可以跳过.

[220]

1. 收缩

至今在表达式中我们假定 d_i 是互异的而且 u_i 非零. 当不是这种情况时，特征方程 $f(\lambda)=0$ 将有 $k<n$ 条垂直渐近线，故有 $k<n$ 个根. 那么，结果变成其余 $n-k$ 个特征值非常便宜地可以得到：若 $d_i=d_{i+1}$，或若 $u_i=0$，可容易地证明 d_i 也是 $\boldsymbol{D}+\rho\boldsymbol{uu}^{\mathrm{T}}$ 的特征值（见问题5.16）. 这个过程称为*收缩*. 实际上或者当 d_i 足够靠近 d_{i+1} 或者当 u_i 足够小时，我们使用一个阈值并收缩 d_i.

实际上，十分频繁地发生收缩：在具有均匀分布特征值的随机稠密矩阵实验中，最大的 $\boldsymbol{D}+\rho\boldsymbol{uu}^{\mathrm{T}}$ 的特征值收缩超过15%，而在具有几何逼近于0特征值的随机稠密矩阵实验中，收缩超过85%！利用这个性质的优点使算法快速是绝对必要的[59,210].

在收缩中的付出不是使特征方程求解更快，这个代价反正只有 $O(n^2)$. 付出是使算法最后步中矩阵乘法快速. 若 $u_i=0$，则相应的特征向量是单位阵的第 i 列 $\boldsymbol{e}_i$（见问题5.16）. 这意味着 $\boldsymbol{Q}'$ 的第 i 列是 $\boldsymbol{e}_i$，故在利用 $\boldsymbol{Q}_1$ 和 $\boldsymbol{Q}_2$ 的两个乘法中计算 $\boldsymbol{Q}$ 的第 i 列不需做什么工作. 当 $d_i=d_{i+1}$ 时存在一个类似的简化. 当许多特征值收缩时，能减少矩阵乘法中的许多运算. 这是在5.3.6节中提出的数值实验中证明的.

2. 解特征方程

当试图用牛顿法求解特征方程时产生一个问题，某些 u_i 是小的但对收缩而言太大. 记得更新 $f(\lambda)=0$ 近似解 λ_j 的牛顿法的原理是：

1. 在接近 $\lambda=\lambda_j$ 处用一个线性函数 $l(\lambda)$ 去逼近 $f(\lambda)$，$l(\lambda)$ 的图像是直线，它与 $f(\lambda)$ 的图像在 $\lambda=\lambda_j$ 处相切.

2. 设 λ_{j+1} 是这个线性近似的零点：$l(\lambda_{j+1})=0$.

图5-2中的图像使用牛顿法不会有明显的困难，因为用靠近每个零点的直线似乎是函数 $f(\lambda)$ 相当好的近似. 现在考察图5-3中的图像，它与图5-2不同仅仅把 u_i^2 从0.5改变为0.001，这对收缩来说远非足够小. $f(\lambda)$ 的图像在左边的图中视觉上不能区分它的垂直渐近线和水平渐近线，故在右边的图中在一条垂直渐近线 $\lambda=2$ 周围把它放大. 我们看出 $f(\lambda)$ 的图像"转角"非常陡并且对多数 λ 值几乎是水平的. 因此，若从几乎任意的 λ_0 开始作牛顿法则线性近似 $l(\lambda)$ 也将几乎是具有微小正斜率的

[221]

水平线，故 λ_1 将是一个巨大的负数，对真正的零点而言这是一个无用的近似.

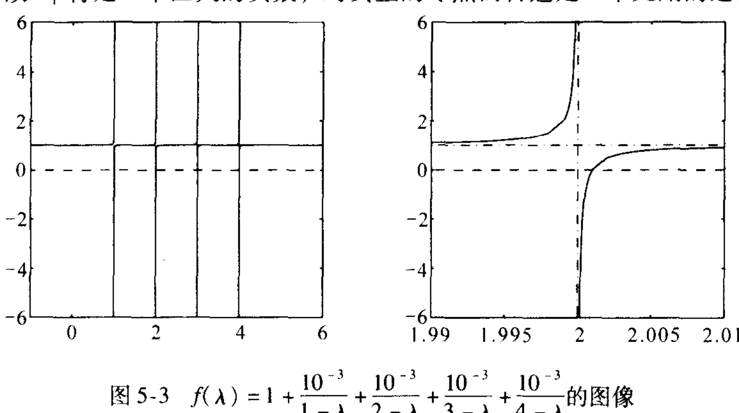


图 5-3 $f(\lambda)=1+\frac{10^{-3}}{1-\lambda}+\frac{10^{-3}}{2-\lambda}+\frac{10^{-3}}{3-\lambda}+\frac{10^{-3}}{4-\lambda}$的图像

可修正牛顿法来处理这种情况如下. 因为用直线 $l(\lambda)$ 近似 $f(\lambda)$ 不是很好，所以用另一个简单函数 $h(\lambda)$ 来近似它. 关于直线没有什么特殊；容易计算并且容易计算零点的任何近似 $h(\lambda)$ 可用于代替牛顿法中的 $l(\lambda)$. 因为 $f(\lambda)$ 在 d_i 和 d_{i+1} 上有极点并且这些极点控制 $f(\lambda)$ 靠近它们的性态，所以很自然当寻求 (d_i,d_{i+1}) 中的根时选择 $h(\lambda)$ 也必须有这些极点，即

$$h(\lambda)=\frac{c_1}{d_i-\lambda}+\frac{c_2}{d_{i+1}-\lambda}+c_3.$$

有几种方式选择常数 c_1，c_2 和 c_3 使 $h(\lambda)$ 近似于 $f(\lambda)$；我们提出一种 LAPACK 程序 slaed4 中使用的一种稍微简化的形式[172,45]. 暂时假定已经选择 c_1，c_2 和 c_3，通过求解等价的二次方程

$$c_1(d_{i+1}-\lambda)+c_2(d_i-\lambda)+c_3(d_i-\lambda)(d_{i+1}-\lambda)=0.$$

可以容易地求使 $h(\lambda)=0$ 的 λ. 给定近似的零点 λ_j，下面是对靠近 λ_j 的 λ 如何计算 c_1，c_2 和 c_3 使得

$$\frac{c_1}{d_i-\lambda}+\frac{c_2}{d_{i+1}-\lambda}+c_3=h(\lambda)\approx f(\lambda)=1+\rho\sum_{k=1}^{n}\frac{u_k^2}{d_k-\lambda}.$$

记

$$f(\lambda)=1+\sum_{k=1}^{i}\frac{u_k^2}{d_k-\lambda}+\sum_{k=i+1}^{n}\frac{u_k^2}{d_k-\lambda}\equiv 1+\Psi_1(\lambda)+\Psi_2(\lambda).$$

222

对 $\lambda\in(d_i,d_{i+1})$，$\Psi_1(\lambda)$ 是正项的和而 $\Psi_2(\lambda)$ 是负项的和. 因此 $\Psi_1(\lambda)$ 和 $\Psi_2(\lambda)$ 都可以准确地计算，然而把它们加在一起很可能会导致对消并在和中失去相对精度. 现在选择 c_1 和 $\hat{c}_1$ 使得

$$h_1(\lambda)\equiv\hat{c}_1+\frac{c_1}{d_i-\lambda}\quad 满足$$

$$h_1(\lambda_j)=\Psi_1(\lambda_j)\text{和}h'_1(\lambda_j)=\Psi'_1(\lambda_j). \tag{5.15}$$

这意味着 $h_1(\lambda)$ 的图像(双曲线)在 $\lambda=\lambda_j$ 上与 $\Psi_1(\lambda)$ 的图像相切. (5.15)式中的两个条件在牛顿法中是常见的条件，我们使用双曲线代替使用直线近似. 容易验证 $c_1=\Psi'_1(\lambda_j)(d_i-\lambda_j)^2$ 和 $\hat{c}_1=\Psi_1(\lambda_j)-\Psi_1'(\lambda_j)(d_i-d_j)$. (见问题5.17)

类似地，选择 c_2 和 $\hat{c}_2$ 使得

$$h_2(\lambda)\equiv\hat{c}_2+\frac{c_2}{d_{i+1}-\lambda}\text{满足}$$

$$h_2(\lambda_j)=\Psi_2(\lambda_j)\text{和}h'_2(\lambda_j)=\Psi'_2(\lambda_j). \tag{5.16}$$

最后,取

$$\begin{aligned}h(\lambda)&=1+h_1(\lambda)+h_2(\lambda)\\&=(1+\hat{c}_1+\hat{c}_2)+\frac{c_1}{d_i-\lambda}+\frac{c_2}{d_{i+1}-\lambda}\\&\equiv c_3+\frac{c_1}{d_i-\lambda}+\frac{c_2}{d_{i+1}-\lambda}.\end{aligned}$$

例5.8　例如，在图5.3的例中，若用 $\lambda_0=2.5$ 开始，则

$$h(\lambda)=\frac{1.1111\cdot10^{-3}}{2-\lambda}+\frac{1.1111\cdot10^{-3}}{3-\lambda}+1,$$

它的图像视觉上可区别于在右图中的 $f(\lambda)$ 的图像. 解 $h(\lambda_1)=0$ 可得 $\lambda_1=2.0011$，它准确到4位十进数字. 继续执行计算，λ_2 准确到11位数字而 λ_3 准确到全部16位数字.　◇

LAPACK 程序 slaed4 中使用的算法是这里描述的算法的一个微小的变形(这里的算法在[172]中称为*中间的方法*(the Middle Way)). LAPACK 程序每个特征值平均二到三次迭代收敛到完全的机器精度，在大量的数值试验中决不多于七步迭代.

223

3. 稳定地计算特征向量

一旦解出特征方程得到 $\boldsymbol{D}+\rho\boldsymbol{uu}^T$ 的特征值 α_i，引理5.2就提供了一个特征向量的简单公式：$(\boldsymbol{D}-\alpha_i\boldsymbol{I})^{-1}\boldsymbol{u}$. 遗憾的是，特别当两个特征值 α_i 和 α_{i+1} 非常接近时，公式可能是不稳定的[59,90,234]. 直觉上，问题是 $(\boldsymbol{D}-\alpha_i\boldsymbol{I})^{-1}\boldsymbol{u}$ 和 $(\boldsymbol{D}-\alpha_{i+1}\boldsymbol{I})^{-1}\boldsymbol{u}$ 是"非常接近的"公式还假定得到正交的特征向量. 更精确地说，当 α_i 和 α_{i+1} 非常接近时，它们也必须靠近 d_i. 所以，当计算 $d_i-\alpha_i$ 和 $d_i-\alpha_{i+1}$ 或者当牛顿迭代期间计算特征方程时，存在一个巨大的对消. 任何一种方法，$d_i-\alpha_i$ 和 $d_i-\alpha_{i+1}$ 都可能含有大的相对误差，因而计算的特征向量 $(\boldsymbol{D}-\alpha_i\boldsymbol{I})^{-1}u$ 和 $(\boldsymbol{D}-\alpha_{i+1}\boldsymbol{I})^{-1}u$ 是十分不准确的且完全不正交的.

为高精度求解特征方程使得 $d_i-\alpha_i$ 和 $d_i-\alpha_{i+1}$ 可高精度算出，早期尝试用双精度算术运算(当输入数据为单精度时)处理这个问题[90,234]. 但当输入数据已经是双精度时，这意味需要四倍精度，而这在许多机器和语言中是不可利用的或者至少是不便宜的. 正如1.5节中描述的，可能利用双精度去模拟四倍精度[234,204]. 只要基本浮点算术运算充分精确地舍入，就可以方便地和相对有效地实行. 特别地，这些模拟

需要 $fl(a \pm b) = (a \pm b)(1+\delta), |\delta| = O(\varepsilon)$，不包括上溢和下溢(见 1.5 节和问题 1.18). 遗憾地，Cray2，YMP 和 C90 舍入精度不够，以致无法使用这些有效的算法.

最后，寻找另一个公式使它成为不需要模拟高精度算术运算. 它是基于下列 Löwner 定理[129,179].

定理 5.10 Löwner. 设 $D = \mathrm{diag}(d_1, \cdots, d_n)$ 是对角阵，$d_n < \cdots < d_1$. 设 $\alpha_n < \cdots < \alpha_1$ 给定，满足交错性质

$$d_n < \alpha_n < \cdots < d_{i+1} < \alpha_{i+1} < d_i < \alpha_i < \cdots < d_1 < \alpha_1.$$

则存在一个向量 $\hat{u}$ 使得 α_i 是 $\hat{D} \equiv D + \hat{u}\hat{u}^{\mathrm{T}}$ 的精确特征值. $\hat{u}$ 的元素向下式给出

$$|\hat{u}_i| = \left[\frac{\prod_{j=1}^{n} (\alpha_j - d_i)}{\prod_{j=1, j \neq i}^{n} (d_j - d_i)} \right]^{1/2}.$$

224

证明 $\hat{D}$ 的特征多项式可同时写成 $\det(\hat{D} - \lambda I) = \prod_{j=1}^{n} (\alpha_j - \lambda)$ 以及(利用(5.13)式和(5.14)式)写成

$$\begin{aligned} \det(\hat{D} - \lambda I) &= \left[\prod_{j=1}^{n} (d_j - \lambda) \right] \cdot \left(1 + \sum_{j=1}^{n} \frac{\hat{u}_j^2}{d_j - \lambda} \right) \\ &= \left[\prod_{j=1}^{n} (d_j - \lambda) \right] \cdot \left(1 + \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\hat{u}_j^2}{d_j - \lambda} \right) + \left[\prod_{\substack{j=1 \\ j \neq i}}^{n} (d_j - \lambda) \right] \cdot \hat{u}_i^2. \end{aligned}$$

置 $\lambda = d_i$，因为 $\det(\hat{D} - \lambda I)$ 的两个表达式相等得到

$$\prod_{j=1}^{n} (\alpha_j - d_i) = \hat{u}_i^2 \cdot \prod_{\substack{j=1 \\ j \neq i}}^{n} (d_j - d_i)$$

或

$$\hat{u}_i^2 = \frac{\prod_{j=1}^{n} (\alpha_j - d_i)}{\prod_{j=1, j \neq i}^{n} (d_j - d_i)}.$$

利用交错性质，可以证明右边的分式是正的，因此可取它的平方根得到所要求的 $\hat{u}_i$ 的表达式. □

下面是计算特征值和特征向量的稳定算法(为简单起见假定 $\rho = 1$).

> **算法 5.3** 计算 $D + uu^{\mathrm{T}}$ 的特征值和特征向量.
>
> 解特征方程 $1 + \sum_{i=1}^{n} \frac{u_i^2}{d_i - \lambda} = 0$ 得到 $D + uu^{\mathrm{T}}$ 的特征值 α_i
>
> 利用 Löwner 定理计算 $\hat{u}$，因此 α_i 是 $D + \hat{u}\hat{u}^{\mathrm{T}}$ 的"精确的"特征值.
>
> 利用引理 5.2 计算 $D + \hat{u}\hat{u}^{\mathrm{T}}$ 的特征向量.

下面简单描述为什么这个算法是数值稳定的. 通过分析特征方程解算器[1]中的停止准则, 可以证明$\|uu^T - \hat{u}\hat{u}^T\|_2 \leq O(\varepsilon)(\|D\|_2 + \|uu^T\|_2)$; 这意味着$D + uu^T$和$D + \hat{u}\hat{u}^T$非常靠近, 可以把$D + \hat{u}\hat{u}^T$的特征值和特征向量看作$D + uu^T$的特征值和特征向量稳定的近似. 接着注意 Löwner 定理中$\hat{u}_i$的公式只需要浮点数$d_j - d_i$和$\alpha_j - d_i$的差, 这些差的乘积和商, 以及一个平方根. 倘若浮点算术运算是足够准确的, 以致对一切$\odot \in \{+, -, \times, /\}$ $\mathrm{fl}(a \odot b) = (a \odot b)(1 + \delta)$, 以及$\mathrm{sqrt}(a) = \sqrt{a}(1 + \delta)$和$|\delta| = O(\varepsilon)$, 那么这个公式可计算到很高的相对精度. 特别地, 不计上溢和下溢可以容易证明

$$\mathrm{fl}\left(\left[\frac{\prod_{j=1}^{n}(\alpha_j - d_i)}{\prod_{j=1, j\neq i}^{n}(d_j - d_i)}\right]^{1/2}\right) = (1 + (4n - 2)\delta) \cdot \left[\frac{\prod_{j=1}^{n}(\alpha_j - d_i)}{\prod_{j=1, j\neq i}^{n}(d_j - d_i)}\right]^{1/2}$$

和$|\delta| = O(\varepsilon)$. 类似地, 引理5.2中的公式也可以计算到高的相对精度, 故可以计算$D + \hat{u}\hat{u}^T$的特征向量到高的相对精度. 特别地, 它们是非常精确地正交的.

总括地说, 倘若浮点算术运算足够精确的话, 算法5.3计算矩阵$D + \hat{u}\hat{u}^T$的非常精确的特征值和特征向量, 它们与原矩阵$D + uu^T$的特征值和特征向量只有一点点不同. 这意味着它是数值稳定的.

读者应该注意, 我们需要充分精确的浮点算术运算是由于在某些 Cray 机器上操作, 明确地防止在[234,204]中提出的四倍精度模拟. 故我们还没有成功地提供一个在这些机器上可靠地工作的算法. 一个额外的诀窍是必要的: 在某些 Cray 机器上不能达到足够准确的运算仅仅是加法和减法, 因为在浮点硬件中没有所谓"保护数位(guard digit)". 这意味着一个操作数的最末位即使它是1, 在加法和减法期间可能被处理为0. 若大部分较高次位删去, 这个"丢失的位"变成重要的. 例如, 从下一个较小的浮点数中减去1, 此时所有前面的位删去, 在 Cray90 上产生过分大的两倍数而在 Cray2 上产生0. 但是若末位已经为0, 则操作是安全的. 因此决窍是在应用 Löwner 定理或算法5.3中的引理5.2之前故意地置d_i的所有末位为0. 这种修正只在d_i和α_i中引起一个小的相对改变, 故算法仍然是稳定的.[2]

在[129,131]中更详尽地描述这个算法, 并且在LAPACK程序`slaed3`中实施这

---

1. 更详细地, 为了达到这个精度特征方程解算器必须求解$\alpha_i - d_i$或$d_{i+1} - \alpha_i$(无论哪个较小)而不是求解α_i.
2. 为了在一台 Cray 机器上置浮点数β的末位为零, 我们可以证明只要置$\beta := (\beta + \beta) - \beta$. 这个花费不多的计算在一台机器上对精确的算术运算完全不改变β(不计容易避免的上溢). 但是在一台 Cray 机器上它置末位为零. 熟悉 Cray 算术运算的读者请证明这个结论. 余下的困难只是完整地移去这行代码保持最佳的编译, 这是某些热衷于优化的人士可能会做的工作; 这通过用一个函数调用存放在主程序中一个单独的文件中的函数实现计算$(\beta + \beta)$. 我们希望即使这种情况在编译变得灵巧足以优化时, Cray 算术运算将消失.

个算法.

226

4. 利用 FMM 加速分而治之

最初发明 FMM[124]是为了计算 n 个电负荷粒子相互作用力或 n 个质量上相互作用重力这些完全不同的问题. 我们仅概述这些问题与求特征值和特征向量的关系,细节留至[131].

设 d_1 到 d_n 是具有电荷 $z_i \cdot u_i$ 的 n 个粒子的三维位置向量. 设 α_1 到 α_n 是具有单位正电荷的另外 n 个粒子的位置向量. 则平方反比律告诉我们由于 d_1 到 d_n 处的粒子作用在粒子 α_j 上的力是和

$$f_j = \sum_{i=1}^{n} \frac{z_i u_i (d_i - \alpha_j)}{\|d_i - \alpha_j\|_2^3}.$$

成比例的. 若我们按二维而不是三维模拟静电,则力定律改变为一次幂反比定律(inverse-first-power law)[1]

$$f_j = \sum_{i=1}^{n} \frac{z_i u_i (d_i - \alpha_j)}{\|d_i - \alpha_j\|_2^2}.$$

因为 d_i 和 α_j 是$\mathbb{R}^2$ 中的向量,所以可把它们考虑为复变量. 此时

$$f_j = \sum_{i=1}^{n} \frac{z_i u_i}{\bar{d}_i - \alpha_j},$$

其中 $\bar{d}_i$ 和 α_j 分别是 d_i 和 α_j 的复共轭. 若 d_i 和 α_j 碰巧为实数,则上式简化为

$$f_j = \sum_{i=1}^{n} \frac{z_i u_i}{d_i - \alpha_j}.$$

现考虑执行一个矩阵 - 向量乘法 $\boldsymbol{f}^{\mathrm{T}} = \boldsymbol{z}^{\mathrm{T}}\boldsymbol{Q}'$, 其中 $\boldsymbol{Q}'$ 是 $\boldsymbol{D} + \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ 的特征向量矩阵. 由引理 5.2, $\boldsymbol{Q}'_{ij} = u_i s_j / (d_i - \alpha_j)$, 其中选择 s_j 为一个标量因子使第 j 列为一个单位向量. 因而 $\boldsymbol{f}^{\mathrm{T}} = \boldsymbol{z}^{\mathrm{T}}\boldsymbol{Q}'$ 的第 j 个元素是

$$f_j = \sum_{i=1}^{n} z_i \boldsymbol{Q}'_{ij} = s_j \sum_{i=1}^{n} \frac{z_i u_i}{d_i - \alpha_j},$$

除了标量因子 s_j 外,这如静电力那样具有相同的和式. 因此,分而治之算法最花费的部分是算法 5.2 最后行中的矩阵乘法,这等价于计算静电力.

227

对 $j = 1, \cdots, n$ 计算这个和式似乎需要 $O(n^2)$ flops. 可用 FMM 和另一些类似于它的算法[124,23]近似地(但非常精确地)以 $O(n \cdot \log n)$ 次 (或甚至 $O(n)$ 次) 替代 $O(n^2)$ 计算这个和式. (见关于"N 体问题的高等级方法"的文献,细节在 PARALLEL_HOMEPAGE 上).

但是单独的这个思想不足以把分而治之代价减少到 $O(n \cdot \log^p n)$. 毕竟,输入特征向量矩阵 $\boldsymbol{Q}$ 有 n^2 个元素,这似乎意味着复杂性至少应该是 n^2. 故我们必须利用少于 n^2 个独立数字来表示 $\boldsymbol{Q}$. 这是可能的,因为 $n \times n$ 阶三对角阵只有 $2n-1$ 个"自由度"(对角线和上对角线上的元素),其中 n 个可用特征值表示,剩余 $n-1$ 个为正交

1. 技术上,这意味着势函数满足两个空间坐标而不是三个空间坐标的泊松方程.

阵 $\boldsymbol{Q}$. 换言之，并非每个正交阵都能是对称三对角阵 $\boldsymbol{T}$ 的特征向量矩阵；只有正交阵的整个 $(n(n-1)/2)$-维集合的 $(n-1)$-维子集能是这种特征向量子阵.

我们将利用由算法5.2计算的分而治之树来表示 $\boldsymbol{Q}$. 与其累计 $\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{Q}_1 & 0\\ 0 & \boldsymbol{Q}_2\end{bmatrix}\cdot\boldsymbol{Q}'$，倒不如在树的每个节点上存放所有的 $\boldsymbol{Q}'$矩阵. 并且将不明显地存放 $\boldsymbol{Q}'$，宁可存放 $\boldsymbol{D},\rho,\boldsymbol{u}$ 以及 $\boldsymbol{D}+\rho\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ 的特征值 α_i. 之所以这样做是因为这一切是使用FMM用 $\boldsymbol{Q}'$ 相乘所需要的. 这就把 $\boldsymbol{Q}$ 所需要的存储量从 n^2 减少到 $O(n\cdot\log n)$. 因此算法的输出是树的节点上所有 $\boldsymbol{Q}'$因子组成的 $\boldsymbol{Q}$ 的“分解因子”形式. 这是 $\boldsymbol{Q}$ 的一个恰当的表示式，因为我们可以使用FMM以 $O(n\cdot\log^p n)$ 次用 $\boldsymbol{Q}$ 乘任何向量.

5.3.4 对分法和逆迭代

对分算法使用西尔维斯特惯性定理(定理5.3)只求我们所要的那些 k 个特征值，代价为 $O(nk)$. 记得 $\mathrm{Inertia}(\boldsymbol{A})=(\nu,\zeta,\pi)$，其中 ν,ζ 和 π 分别是 $\boldsymbol{A}$ 的负特征值，零特征值和正特征值. 假设 X 非奇异，西尔维斯特惯性定理断言 $\mathrm{Inertia}(\boldsymbol{A})=\mathrm{Inertia}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X})$.

现在假设使用高斯消元法分解 $\boldsymbol{A}-z\boldsymbol{I}=\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$，其中 $\boldsymbol{L}$ 非奇异而 $\boldsymbol{D}$ 为对角阵. 则 $\mathrm{Inertia}(\boldsymbol{A}-z\boldsymbol{I})=\mathrm{Inertia}(\boldsymbol{D})$. 因为 $\boldsymbol{D}$ 为对角阵，所以它的惯性计算是平凡的(在下文中利用诸如“$\#d_{ii}<0$”的记号表示“d_{ii}的值小于零的个数”)

$$
\begin{aligned}
\mathrm{Inertia}(\boldsymbol{A}-z\boldsymbol{I}) &= (\#d_{ii}<0,\#d_{ii}=0,\#d_{ii}>0)\\
&= (\#\boldsymbol{A}-z\boldsymbol{I}\text{ 的负特征值},\\
&\qquad \#\boldsymbol{A}-z\boldsymbol{I}\text{ 的零特征值},\\
&\qquad \#\boldsymbol{A}-z\boldsymbol{I}\text{ 的正特征值})\\
&= (\#\boldsymbol{A}\text{ 的特征值}<z,\\
&\qquad \#\boldsymbol{A}\text{ 的特征值}=z,\\
&\qquad \#\boldsymbol{A}\text{ 的特征值}>z).
\end{aligned}
$$

228

假设 $z_1<z_2$，并计算 $\mathrm{Inertia}(\boldsymbol{A}-z_1\boldsymbol{I})$ 和 $\mathrm{Inertia}(\boldsymbol{A}-z_2\boldsymbol{I})$. 则区间 $[z_1,z_2)$ 内的特征值个数等于 $(\#\boldsymbol{A}$ 的特征值 $<z_2)-(\#\boldsymbol{A}$ 的特征值 $<z_1)$.

为把这个观察安排到一个算法中，定义

$$\mathrm{Negcount}(\boldsymbol{A},z)=\#\boldsymbol{A}\text{ 的特征值}<z$$

算法5.4　对分法：求 $[a,b)$ 内的 $\boldsymbol{A}$ 的所有特征值至给定的误差容限tol：

$n_a=\mathrm{Negcount}(\boldsymbol{A},a)$

$n_b=\mathrm{Negcount}(\boldsymbol{A},b)$

if $n_a=n_b$, quit…因为在 $[a,b)$ 中不存在特征值.

put$[a,n_a,b,n_b]$ onto *Worklist*

/* *Worklist* 包含一个含有 $n-n_a$ 到 $n-n_b+1$ 个特征值的区间 $[a,b)$ 目录，对这些区间算法将反复对分直到它们窄于tol. */

（续）

```
while Worklist 非空 do
从 Worklist 删掉[low,n_low,up,n_up]
if (up - low < tol)  then
  print "在[low,up]中有 n_up - n_low 个特征值"
else
    mid = (low + up)/2
    n_mid = Negcount(A,mid)
    if n_mid > n_low then… 在 [low,mid) 中有特征值
      put[low,n_low,mid,n_mid] onto Worklist
    end if
    if n_up > n_mid then…在 [mid,up) 中有特征值
      put[mid,n_mid,up,n_up] onto Worklist
    end if
end if
end while
```

若 $\alpha_1 \geqslant \cdots \geqslant \alpha_n$ 是特征值，则同样的思想可用于计算 α_j，对 $j=j_0, j_0+1, \cdots, j_1$. 这是因为我们知道 $\alpha_{n-n_{\text{low}}}$ 到 $\alpha_{n-n_{\text{up}}+1}$ 位于区间 [low, up) 中.

若 A 稠密，则可用2.7.2节中描述的选主元的对称高斯消元法执行 Negcount(A,z). 但这样做每个计算将花费 $O(n^3)$ flops，故这不是有效的代价. 另一方面，倘若不选主元对三对角阵 $\boldsymbol{A}$ 计算 Negcount$(\boldsymbol{A},z)$ 是十分简单的：

$$\boldsymbol{A}-z\boldsymbol{I} \equiv \begin{bmatrix} a_1-z & b_1 & & \\ b_1 & a_2-z & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n \end{bmatrix} = \boldsymbol{LDL}^{\mathrm{T}}$$

229

$$\equiv \begin{bmatrix} 1 & & & \\ l_1 & \ddots & & \\ & \ddots & \ddots & \\ & & l_{n-1} & 1 \end{bmatrix} \cdot \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \cdot \begin{bmatrix} 1 & l_1 & & \\ & \ddots & \ddots & \\ & & \ddots & l_{n-1} \\ & & & 1 \end{bmatrix},$$

故 $a_1-z=d_1, d_1 l_1=b_1$，之后 $l_{i-1}^2 d_{i-1}+d_i=a_i-z, d_i l_i=b_i$. 把 $l_i=b_i/d_i$ 代入到 $l_{i-1}^2 d_{i-1}+d_i=a_i-z$ 中得到简单的递推

$$d_i=(a_i-z)-\frac{b_{i-1}^2}{d_{i-1}}. \tag{5.17}$$

注意我们不选主元，故特别当 d_i 小的时候，你可能认为这是很不稳定的. 事实上，因为 $A-zI$ 三对角，所以可以证明(5.17)是非常稳定的[73,74,156].

引理5.3　利用(5.17)的浮点算术运算计算的 d_i 如同从 $\hat{A}$ 精确地计算 $\hat{d}_i$ 有相同的符号(因而计算相同的惯性)，这里 $\hat{A}$ 非常接近于 A：

$$(\hat{A})_{ii}\equiv\hat{a}_i=a_i,(\hat{A})_{i,i+1}\equiv\hat{b}_i=b_i(1+\varepsilon_i),\text{其中}|\varepsilon_i|\leq 2.5\varepsilon+O(\varepsilon^2).$$

证明　设 $\tilde{d}_i$表示利用(5.17)式算出的量，包含舍入误差：

$$\tilde{d}_i=\left[(a_i-z)(1+\varepsilon_{-,1,i})-\frac{b_{i-1}^2(1+\varepsilon_{*,i})}{\tilde{d}_{i-1}}\cdot(1+\varepsilon_{/,i})\right](1+\varepsilon_{-,2,i}), \tag{5.18}$$

其中所有的 ε 数量上以机器舍入 ε 为界，而它们的下标指出它们来自于哪种浮点运算(例如 $\varepsilon_{-,2,i}$是来自计算 $\tilde{d}_i$的第2个减法). 定义新的变量

$$\hat{d}_i=\frac{\tilde{d}_i}{(1+\varepsilon_{-,1,i})(1+\varepsilon_{-,2,i})}, \tag{5.19}$$

$$\hat{b}_{i-1}=b_{i-1}\left[\frac{(1+\varepsilon_{*,i})(1+\varepsilon_{/,i})}{(1+\varepsilon_{-,1,i})(1+\varepsilon_{-,1,i-1})(1+\varepsilon_{-,2,i-1})}\right]^{1/2}$$

$$\equiv b_{i-1}(1+\varepsilon_i).$$

注意 $\hat{d}_i$ 和 $\tilde{d}_i$具有相同的符号，而 $|\varepsilon_i|\leq 2.5\varepsilon+O(\varepsilon^2)$. 把(5.19)代入到(5.18)得到

$$\hat{d}_i=a_i-z-\frac{\hat{b}_{i-1}^2}{\hat{d}_{i-1}},$$

证毕. □

完整的分析必须考虑上溢或下溢的可能性，实际上利用IEEE算术运算的例外处理工具，即使当某些 d_{i-1}恰好为零我们也能安全地计算！对此情况 $d_i=-\infty$，$d_{i+1}=a_{i+1}-z$ 并且计算无例外地继续下去[73,81]. [230]

对一个三对角阵单独调用Negcount的代价至多为 $4n$ flops. 所以求 k 个特征值的全部代价为 $O(kn)$. 这由LAPACK中的程序sstebz来实施.

注意对分法线性收敛，对区间的每次对分精确的数位超过一位. 有许多加速收敛的方法，利用牛顿法及其相关的算法求特征多项式的零点(它可以通过把所有的 d_i 一起相乘来计算)[173,174,175,176,178,269].

一旦已算出(挑选的)特征值，为计算特征向量，可以使用逆迭代(算法4.2)；这可利用LAPACK程序sstein. 因为可利用准确的特征值作为位移，所以通常作1到2次迭代就收敛. 此时每个特征向量代价是 $O(n)$ flops，因为逆迭代的一步只

需要求解一个三对角方程组(见 2.7.3 节). 当几个计算的特征值 $\hat{\alpha}_i,\cdots,\hat{\alpha}_j$ 紧靠在一起时，它们相应的特征向量 $\hat{q}_i,\cdots,\hat{q}_j$ 可能不正交. 此时再正交(reorthogonalize)计算的特征向量的算法计算 QR 分解 $[\hat{q}_i,\cdots,\hat{q}_j]=QR$ 并且用 Q 的第 k 列替代每个 $\hat{q}_k$；这就保证每个 $\hat{q}_k$ 规范正交. 通常利用 MGS 正交化过程(算法 3.1)计算这个 QR 分解；即每个计算的特征向量具有任意的明确地减去前面计算的特征向量的方向中的分量. 当串的数量 k 小的时候，这个再正交的代价 $O(k^2n)$ 是小的，故原则上所有的特征值和所有的特征向量可用对分法接着用逆迭代计算总共不过 $O(n^2)$ flops. 这比 QR 迭代或分而治之(最坏情况)的代价 $O(n^3)$ 快得多. 得到这个可靠地加速的障碍是当串的数量 k 大的时候，即是 n 的一个相当大的分数，则总代价再次增长到 $O(n^3)$. 更坏地，不保证计算的特征向量准确或正交(麻烦是再正交一组几乎相关的 $\hat{q}_k$ 之后，对消可能意味着某些计算的特征向量由一些比舍入误差小的向量组成.)

然而，这个问题有新的进展[105,83,201,203]，并且现在它似乎可能“修补”逆迭代以提供准确的、正交的特征向量，每个特征向量花费不超过 $O(n)$ flops. 这将使对分法和“修补的”逆迭代成为在所有情况下可选的算法，不论要求多少特征值和特征向量. 我们期待在未来的版本中描述这个算法.

注意对分法和逆迭代是“令人为难地并行的”，因为每个特征值和随后的特征向量可能独立于其他的特征值和特征向量求出. (这足以推定逆迭代已被修补以致对众多其他的特征向量再正交化不再必要.)这就使这些算法对并行计算机非常有吸引力. [231]

5.3.5 雅可比法

雅可比法不像前面的方法所做的那样开始先将 A 化为三对角形式，而改为对原来的稠密阵操作. 雅可比法通常比前面的方法慢得多，而兴趣得以维持仅仅是因为它有时能比前面的方法具有较高的精度计算微小的特征值和特征向量，而且它能容易地并行计算. 下面只描述雅可比法的基本执行过程并把高精度的讨论推迟到 5.4.3 节.

给定一个对称阵 $A=A_0$，雅可比法产生一系列正交相似矩阵 $A_1,A_2,\cdots$，它们最终收敛于一个特征值在对角线上的对角阵. A_{i+1} 由 A_i 通过公式 $A_{i+1}=J_i^{\mathrm{T}}A_iJ_i$ 得到，其中 J_i 是一个称为雅可比旋转的正交阵. 因此

$$\begin{aligned}A_m&=J_{m-1}^{\mathrm{T}}A_{m-1}J_{m-1}\\&=J_{m-1}^{\mathrm{T}}J_{m-2}^{\mathrm{T}}A_{m-2}J_{m-2}J_{m-1}=\cdots\\&=J_{m-1}^{\mathrm{T}}\cdots J_0^{\mathrm{T}}A_0J_0\cdots J_{m-1}\\&\equiv J^{\mathrm{T}}AJ.\end{aligned}$$

若适当地选取每个 J_i，对大的 m，A_m 逼近于对角阵 Λ. 因此可记 $\Lambda\approx J^{\mathrm{T}}AJ$ 或

$\boldsymbol{J\Lambda J}^{\mathrm{T}} \approx \boldsymbol{A}$. 所以 $\boldsymbol{J}$ 的列是近似的特征向量.

我们将反复地选取 $\boldsymbol{J}_i$ 使 $\boldsymbol{A}_{i+1}=\boldsymbol{J}_i^{\mathrm{T}}\boldsymbol{A}_i\boldsymbol{J}_i$ 的一对非对角元同时为零，使 $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{AJ}$ 变成几乎对角阵. 将通过选取 $\boldsymbol{J}_i$ 为吉文斯旋转来做这项工作

$$\boldsymbol{J}_i=\boldsymbol{R}(j,k,\theta)\equiv \begin{matrix} \\ \\ \\ j \\ \\ k \\ \\ \\ \\ \end{matrix}\begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & \cos\theta & & -\sin\theta & & \\ & & & & \ddots & & & \\ & & & \sin\theta & & \cos\theta & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \\ & & & & & & & & 1 \end{bmatrix},$$

（列标：j 对应 $\cos\theta$、$\sin\theta$ 所在列，k 对应 $-\sin\theta$、$\cos\theta$ 所在列）

其中 θ 的选取是使 $\boldsymbol{A}_{i+1}$ 的 j,k 和 k,j 元素化为零. 为确定 θ(实际上是确定 $\cos\theta$ 和 $\sin\theta$),记

232

$$\begin{bmatrix} a_{jj}^{(i+1)} & a_{jk}^{(i+1)} \\ a_{kj}^{(i+1)} & a_{kk}^{(i+1)} \end{bmatrix}=\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^{\mathrm{T}}\begin{bmatrix} a_{jj}^{(i)} & a_{jk}^{(i)} \\ a_{kj}^{(i)} & a_{kk}^{(i)} \end{bmatrix}\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$=\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

其中 λ_1 和 λ_2 是 $\begin{bmatrix} a_{jj}^{(i)} & a_{jk}^{(i)} \\ a_{kj}^{(i)} & a_{kk}^{(i)} \end{bmatrix}$ 的特征值.

计算 $\cos\theta$ 和 $\sin\theta$ 是容易的：利用对称性，简记 $c\equiv\cos\theta$ 和 $s\equiv\sin\theta$，并且为简单起见删去上标(i)通过相乘最后的表达式得到

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}=\begin{bmatrix} a_{jj}c^2+a_{kk}s^2+2sca_{jk} & sc(a_{kk}-a_{jj})+a_{jk}(c^2-s^2) \\ sc(a_{kk}-a_{jj})+a_{jk}(c^2-s^2) & a_{jj}s^2+a_{kk}c^2-2sca_{jk} \end{bmatrix}.$$

置非对角元为0和求解 θ，我们得到 $0=sc(a_{kk}-a_{jj})+a_{jk}(c^2-s^2)$，或

$$\frac{a_{jj}-a_{kk}}{2a_{jk}}=\frac{c^2-s^2}{2sc}=\frac{\cos2\theta}{\sin2\theta}=\cot2\theta\equiv\tau.$$

今设 $t=\dfrac{s}{c}=\tan\theta$ 并注意 $t^2+2\tau t-1=0$(由二次方程求根公式)得到 $t=\dfrac{\mathrm{sign}(\tau)}{|\tau|+\sqrt{1+\tau^2}}$，$c=\dfrac{1}{\sqrt{1+t^2}}$ 和 $s=t\cdot c$. 我们总结这些推导于下列算法中.

算法 5.5 对 A 的坐标 j, k 计算和应用一次雅可比旋转:

```
proc Jacobi-Rotation(A,j,k)
  if |a_jk| 不是太小
    τ = (a_jj - a_kk)/(2 · a_jk)
    t = sign(τ)/(|τ| + sqrt(1 + τ^2))
    c = 1/sqrt(1 + t^2)
    s = c · t
    A = R^T(j,k,θ) · A · R(j,k,θ) …其中 c = cosθ 和 s = sinθ
    if 要求特征向量
      J = J · R(j,k,θ)
    end if
end if
```

应用 $R(j,k,\theta)$ 于 A (或 J) 的代价仅为 $O(n)$ flops，因为只修正 A 的第 j 行、第 j 列和第 k 行、第 k 列(和 J 的第 j 列和第 k 列). 下面是全部的雅可比算法.

算法 5.6 求对称阵特征值的雅可比法:

```
repeat
  选择一对 j, k
  调用 Jacobi-Rotation(A,j,k)
until(A)充分地对角化
```

233

我们仍需决定如何挑选 j, k 对. 有几种可能性. 为度量收敛的进展以及描述这些可能性，定义

$$\mathrm{off}(A) = \sqrt{\sum_{1\le j<k\le n} a_{jk}^2}.$$

因而 off(A)是 A 的(上面)非对角元的平方和的方根，故 A 为对角阵当且仅当 off $A)=0$. 我们的目标是使 off(A)很快地逼近于 0. 下面的引理告诉我们对每一个雅可比旋转 off(A)都单调地递减.

引理 5.4 对任意的 $j\neq k$，设 A'是执行 Jacobi-Rotation(A,j,k)之后的矩阵. 则 $\mathrm{off}^2(A')=\mathrm{off}^2(A)-a_{jk}^2$.

证明 注意除了第 j 行、第 k 行和第 j 列、第 k 列之外 $A'=A$. 记

$$\mathrm{off}^2(A) = \left(\sum_{\substack{1\le j'<k'\le n\\ j'\neq j\text{或}k'\neq k}} a_{j'k'}^2\right) + a_{jk}^2 \equiv S^2 + a_{jk}^2$$

类似地 $\mathrm{off}^2(A')=S'^2+a'^2_{jk}=S'^2$，因为调用 Jacobi-Rotation$(A,j,k)$之后 $a'_{jk}=0$. 因为

对任意的 X 和任意的正交阵 Q, $\|X\|_F=\|QX\|_F$ 和 $\|X\|_F=\|XQ\|_F$，所以可以证明 $S^2=S'^2$. 于是 $\mathrm{off}^2(A')=\mathrm{off}^2(A)-a_{jk}^2$. □

下面的算法是(雅可比于1846年给出的)原始算法，虽然它太慢以致无法使用，但它具有一个引人注目的分析.

> **算法 5.7**　经典雅可比算法：
> while off(A) > tol（这里 tol 是由用户设置的停止标准）
> 　选择 j 和 k 使 a_{jk} 是数量上最大的非对角元
> 　调用 Jacobi-Rotation(A,j,k)
> end while

定理 5.11　在经典雅可比算法中一次雅可比旋转之后，有 $\mathrm{off}(A')\le\sqrt{1-\frac{1}{N}}\,\mathrm{off}(A)$，其中 $N=\frac{n(n-1)}{2}=A$ 的对角线以上元素个数. k 步 Jacobi-Rotations 之后 off(·)不会超过 $(1-\frac{1}{N})^{k/2}\mathrm{off}(A)$.

234

证明　由引理5.4，一步之后 $\mathrm{off}^2(A')=\mathrm{off}^2(A)-a_{jk}^2$,其中 a_{jk} 是最大的非对角元. 因此 $\mathrm{off}^2(A)\le\frac{n(n-1)}{2}a_{jk}^2$，或 $a_{jk}^2\ge\frac{1}{n(n-1)/2}\mathrm{off}^2(A)$，故 $\mathrm{off}^2(A)-a_{jk}^2\le\left(1-\frac{1}{N}\right)\mathrm{off}^2(A)$. 由此可证结论. □

故经典雅可比算法至少线性收敛，每次至少以因子 $\sqrt{1-\frac{1}{N}}$ 递减误差(用 off(A)度量). 事实上，它最终二次收敛.

定理 5.12　雅可比法 N 步之后(即每次足够步数选择每个 a_{jk})局部二次收敛. 这意味着对足够大的 i

$$\mathrm{off}(A_{i+N})=O(\mathrm{off}^2(A_i)).$$

实际上并不使用经典的雅可比算法，因为最大元的搜索太慢：每次雅可比旋转需要搜索 $\frac{n^2-n}{2}$ 个元素，而执行的代价仅 $O(n)$ flops，故对大的 n，搜索时间将占支配地位. 改为使用下列简单的方法选择 j 和 k.

> **算法 5.8**　行循环雅可比：逐行扫描 A 的非对角元.
> repeat
> 　for $j=1$ to $n-1$
> 　　for $k=j+1$ to n

(续)

```
调用 Jacobi-Rotation(A,j,k)
      end for
   end for
until A 充分地对角化
```

当完整的经过内循环 Jacobi-Rotation(A,j,k) 只选择 $c=1$ 和 $s=0$ 时 A 不再改变. 循环雅可比算法类似于经典雅可比算法也是渐近二次收敛[262,p.270].

一次雅可比"扫描"(其中每对 j, k 选取一次)的代价近似地是化为三对角形式并利用 QR 迭代计算特征值和特征向量的代价的一半, 而多于利用分而治之的代价. 因为雅可比方法常常作 5~10 次扫描收敛, 它比其他有关方法慢得多.

5.3.6 性能比较

本节中分析对称特征问题三个最快速算法的性能: QR 迭代, 带逆迭代的对分法和分而治之. 更多的细节可在[10,第3章]中或 NETLIB/lapack/lug/lapack_lug.html 中找到.

235

我们从讨论最快的算法开始, 之后与其他的算法作比较. 我们利用 LAPACK 程序 ssyevd. 只求特征值的算法是先化为三对角形式随后用 QR 迭代, 运算量为 $\frac{4}{3}n^3+O(n^2)$ flops. 求特征值和特征向量的算法是先化为三对角形式随后用分而治之. 在一台有高峰速度为 266 Mflops 的工作站 IBM RS6000/590 上记录 ssyevd 的时间, 虽然最佳的矩阵乘法对 100×100 阶矩阵只运行 233 Mflops, 对 1000×1000 矩阵运行 256Mflops. 实际的性能在下表中给出. "Mflops rate" 是代码用 Mflops 计算的实际速度, 而 "Time/Time(Matmul)" 是用阶数同样大小的两个方阵相乘时间除求解特征问题的时间. 对足够大的矩阵, 矩阵乘法和只求一个对称阵的特征值耗时几乎相同. (相反, 非对称特征问题最少是 16 倍, 代价更昂贵[10]). 另外求特征向量少许低于矩阵乘法耗时的三倍.

维数	只求特征值		求特征值和特征向量	
	Mflop rate	Time/Time(Matmul)	Mflop rate	Time/Time(Matmul)
100	72	3.1	72	9.3
1000	160	1.1	174	2.8

现在比较 QR 迭代, 带逆迭代的对分法和分而治之的相对性能. 在图 5-4 和图 5-5中这些算法分别记为 QR, BZ(在 LAPACK 中执行对分法的程序为 sstebz)和 DC. 在这些图中水平轴是矩阵维数, 而垂直轴是用 DC 的时间相除的时间, 所以 DC 曲线是在1处的一条水平线, 而其他的曲线度量 BZ 和 QR 比 DC 慢多少倍. 图

5-4只指出三对角特征问题的时间，然而图 5-5 指出从稠密矩阵出发的整个时间.

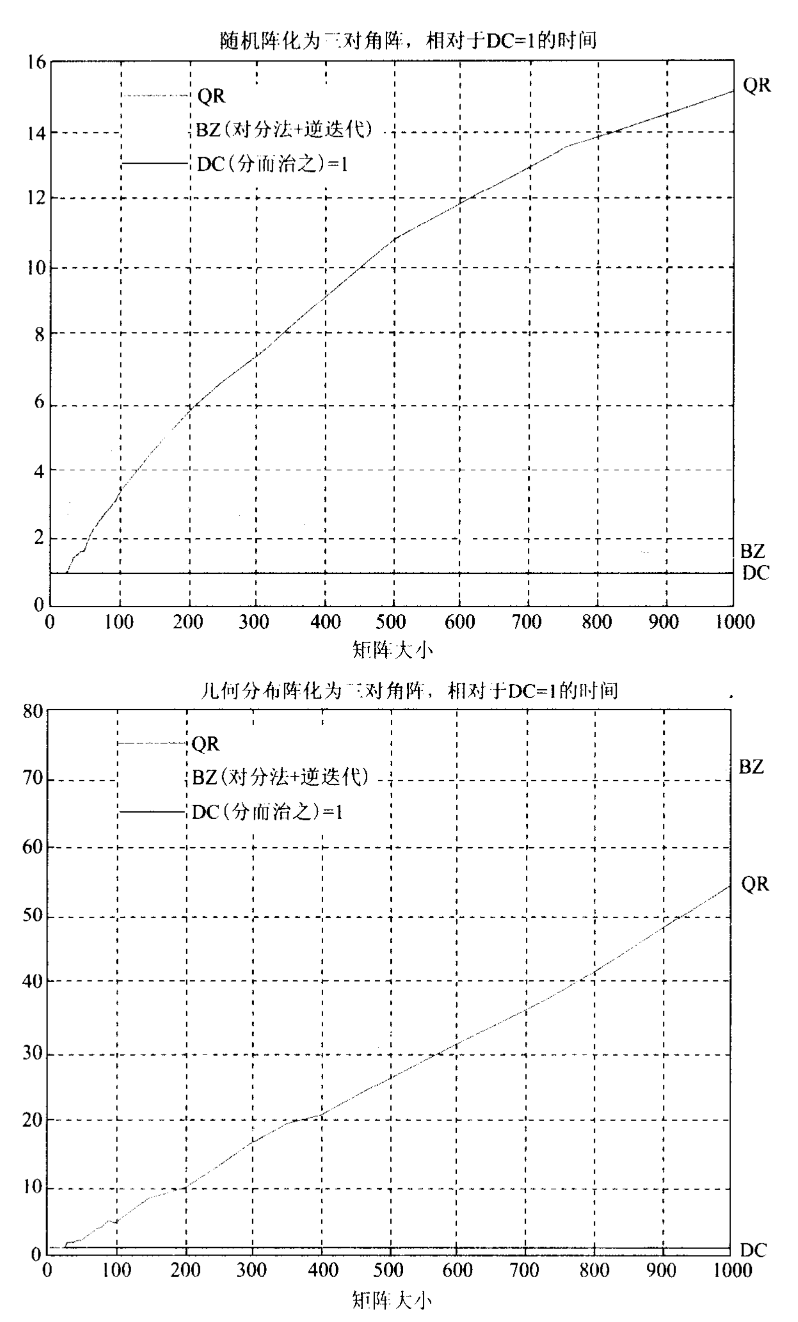


图 5-4　相对于分而治之求对称三对角阵特征值和特征向量的速度

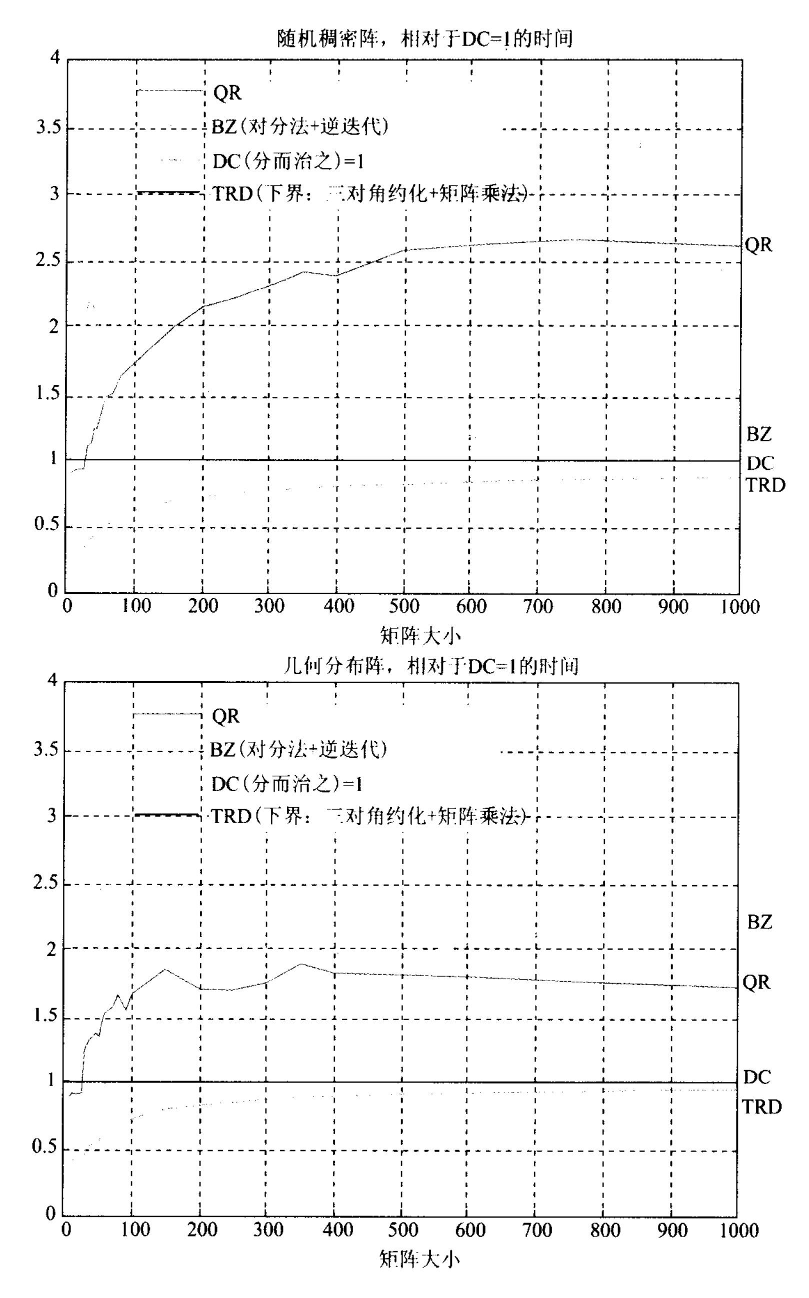


图 5-5　相对于分而治之求对称稠密阵特征值和特征向量的速度

在图 5-5 的顶部图像中测试的矩阵是随机的对称阵；在图 5-4 中三对角阵是约化这些稠密阵为三对角形式所得到的．这样的随机矩阵通常有良好分离的特征值，故

逆迭代需要一点点或不费时的再正交. 所以 BZ 性能与 DC 相当, 即使 QR 相当慢, 对大的矩阵在三对角化阶段直到慢 15 倍.

在底部的两个图像中, 稠密对称阵有特征值 $1, 0.5, 0.25, \cdots, 0.5^{n-1}$. 换言之, 有许多接近于零的成串的特征值, 故逆迭代有许多再正交要做. 因此 BZ 的三对角化部分比 DC 慢 70 倍以上. QR 也比 DC 慢 54 倍, 因为当有一大串特征值时, 由于收缩的缘故, DC 实际上加速了.

QR, BZ 和 DC 之间速度的区别在图5-5 中比图5-4 中更不显而易见, 因为图5-5 包含化为三对角形式以及把三对角矩阵特征值转换成原稠密阵特征值的共同的开销 $O(n^3)$, 这个共同的开销记为 TRD. 因为在图 5-5 中 DC 接近于 TRD, 这意味着 DC 的任何进一步加速对稠密算法的整体速度只会产生很少的差别.

5.4　奇异值分解算法

在定理 3.3 中证明一般矩阵 $\boldsymbol{G}$ 的 SVD 与对称阵 $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$, $\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}}$ 和 $\begin{bmatrix} O & G^{\mathrm{T}} \\ G & O \end{bmatrix}$ 的特征分解是密切相关的. 利用这些事实, 前节中的算法可转换成 SVD 算法. 然而转换不是直接的, 因为常常可以利用 SVD 额外的结构使算法更有效或更准确[120,80,67].

除雅可比法外, 对称阵 $\boldsymbol{A}$ 特征分解的所有算法都具有下列结构:

1. 用正交阵 $\boldsymbol{Q}_1$ 把 $\boldsymbol{A}$ 化为三对角形式 T: $\boldsymbol{A}=\boldsymbol{Q}_1\boldsymbol{T}\boldsymbol{Q}_1^{\mathrm{T}}$.

2. 求 $\boldsymbol{T}$ 的特征分解: $\boldsymbol{T}=\boldsymbol{Q}_2\boldsymbol{\Lambda}\boldsymbol{Q}_2^{\mathrm{T}}$, 其中 $\boldsymbol{\Lambda}$ 是特征值的对角阵而 $\boldsymbol{Q}_2$ 是正交阵, 其列为特征向量.

3. 合并这些分解得到 $\boldsymbol{A}=(\boldsymbol{Q}_1\boldsymbol{Q}_2)\boldsymbol{\Lambda}(\boldsymbol{Q}_1\boldsymbol{Q}_2)^{\mathrm{T}}$. $\boldsymbol{Q}=\boldsymbol{Q}_1\boldsymbol{Q}_2$ 的列是 $\boldsymbol{A}$ 的特征向量.

除了雅可比法外, 一般阵 $\boldsymbol{G}$ 的所有 SVD 算法都具有类似的结构:

1. 用正交阵 $\boldsymbol{U}_1$ 和 $\boldsymbol{V}_1$ 把 $\boldsymbol{G}$ 化为双对角形式 $\boldsymbol{B}$: $\boldsymbol{G}=\boldsymbol{U}_1\boldsymbol{B}\boldsymbol{V}_1^{\mathrm{T}}$. 这意味着 $\boldsymbol{B}$ 仅在主对角线和第一上对角线上非零.

2. 求 $\boldsymbol{B}$ 的 SVD: $\boldsymbol{B}=\boldsymbol{U}_2\Sigma\boldsymbol{V}_2^{\mathrm{T}}$, 其中 Σ 是奇异值的对角阵, 而 $\boldsymbol{U}_2$ 和 $\boldsymbol{V}_2$ 是正交阵, 它们的列分别是左和右奇异向量.

3. 合并这些分解得到 $\boldsymbol{G}=(\boldsymbol{U}_1\boldsymbol{U}_2)\Sigma(\boldsymbol{V}_1\boldsymbol{V}_2)^{\mathrm{T}}$. $\boldsymbol{U}=\boldsymbol{U}_1\boldsymbol{U}_2$ 和 $\boldsymbol{V}=\boldsymbol{V}_1\boldsymbol{V}_2$ 的列分别是 $\boldsymbol{G}$ 的左和右奇异向量.

236 ~ 239

约化为双对角形式是通过 4.4.7 中的算法实现的. 回顾那里的讨论知道计算 $\boldsymbol{B}$ 它花费 $\frac{8}{3}n^3+O(n^2)$ flops; 如果只计算奇异值 Σ, 这就是所需的全部代价. 它花费了另外的 $4n^3+O(n^2)$ flops 计算 $\boldsymbol{U}_1$ 和 $\boldsymbol{V}_1$, 这是一起计算奇异向量所需的代价.

下列简单的引理指出如何把求双对角阵 B 的 SVD 转换成对称三对角阵 T 的特征分解.

引理 5.5　设 $\boldsymbol{B}$ 是具有对角元 $a_1, \cdots, a_n$ 和上对角元 $b_1, \cdots, b_{n-1}$ 的 $n\times n$ 阶双对角阵. 有三种方法把求 $\boldsymbol{B}$ 的 SVD 问题转换成求对称三对角阵的特征值和特征向量.

1. 设 $A=\begin{bmatrix}0 & B^{T}\\ B & 0\end{bmatrix}$. P 是置换矩阵 $P=[e_1,e_{n+1},e_2,e_{n+2},\cdots,e_n,e_{2n}]$，其中 e_i 是 $2n\times 2n$ 阶单位阵的第 i 列. 则 $T_{ps}\equiv P^{T}AP$ 是对称三对角的. 下标“ps”代表完美的洗牌，因为用 P 乘向量 x“搅乱”x 的元素类似于一付纸牌洗牌. 可以证明 T_{ps} 在其主对角线上全为零，而其上对角线和下对角线上为 $a_1,b_1,a_2,b_2\cdots,b_{n-1},a_n$. 若 $T_{ps}x_i=\alpha_i x_i$ 是 T_{ps} 的一个特征对，x_i 为单位向量，则 $\alpha_i=\pm\sigma_i$，其中 σ_i 是 B 的一个奇异值且 $Px_i=\frac{1}{\sqrt{2}}\begin{bmatrix}v_i\\ \pm u_i\end{bmatrix}$，其中 u_i 和 v_i 分别是 B 的左和右奇异向量.

2. 设 $T_{BB^{T}}\equiv BB^{T}$，则 $T_{BB^{T}}$ 是有对角元 $a_1^2+b_1^2,a_2^2+b_2^2,\cdots,a_{n-1}^2+b_{n-1}^2,a_n^2$ 的对称三对角阵，而上对角线和下对角线上的元素为 $a_2b_1,a_3b_2,\cdots,a_nb_{n-1}$. B 的奇异值是 $T_{BB^{T}}$ 的特征值的平方根，并且 B 的左奇异向量是 $T_{BB^{T}}$ 的特征向量.

3. 设 $T_{B^{T}B}\equiv B^{T}B$，则 $T_{B^{T}B}$ 是有对角元 $a_1^2,a_2^2+b_1^2,a_3^2+b_2^2,\cdots,a_n^2+b_{n-1}^2$ 的对称三对角阵，而上对角线和下对角线上的元素为 $a_1b_1,a_2b_2,\cdots,a_{n-1}b_{n-1}$. B 的奇异值是 $T_{B^{T}B}$ 的特征值的平方根，并且 B 的右奇异向量是 $T_{B^{T}B}$ 的特征向量. $T_{B^{T}B}$ 不含有关 B 的左奇异向量的信息.

证明见问题 5.19.

因此，由引理 5.5 原则上可对三种三对角矩阵之中一个任意的应用 QR 迭代，分而治之或带逆迭代的对分法，然后从产生的特征分解中取出奇异值和(也许只要左或右)奇异向量. 然而，这个简单的方法忽视基础的 SVD 问题的特殊性质，将同时牺牲速度和精度. 对此我们给出两个说明.

首先，对 T_{PS} 运行对称三对角 QR 迭代或分而治之将是无效的. 这是因为这些算法同时计算 T_{PS} 的全部特征值(也许还有特征向量)，而引理 5.5 告诉我们，我们只需要非负特征值(也许还有特征向量). 对微小的奇异值的奇异向量也存在某些精确度困难. 240

其次，直接构成 $T_{BB^{T}}$ 或 $T_{B^{T}B}$ 是数值不稳定的. 事实上在 B 的小奇异值中可能丢失一半精度. 例如，设 $\eta=\varepsilon/2$，故 $1+\eta$ 在浮点算术运算中舍入为 1. 设 $B=\begin{bmatrix}1 & 0\\ 1 & \sqrt{\eta}\end{bmatrix}$，它有接近于 $\sqrt{2}$ 和 $\sqrt{\eta/2}$ 的奇异值. 因而 $B^{T}B=\begin{bmatrix}1 & 1\\ 1 & 1+\eta\end{bmatrix}$ 舍入到一个恰好奇异的矩阵 $T_{B^{T}B}=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}$. 因此把 $1+\eta$ 舍入为 1 把较小的计算奇异值从它的接近于 $\sqrt{\eta/2}=\sqrt{\varepsilon}/2$ 的真值变为 0. 相反，一个向后稳定的算法应该改变奇异值不大于 $O(\varepsilon)\|B\|_2=O(\varepsilon)$. 在 IEEE 双精度浮点算术运算中，$\varepsilon\approx 10^{-16}$ 而 $\sqrt{\varepsilon}/2\approx 10^{-8}$，故构成 $B^{T}B$ 引入的误差大于舍入 10^{8} 倍，这是一个极大的改变. 直接构成 $T_{BB^{T}}$ 能够发生丢失相同的精度.

因为不稳定性由计算 T_{BB^T} 或 T_{B^TB} 所引起的，所以好的 SVD 算法直接对 B 或可能对 T_{ps} 操作.

下面扼要地描述计算 SVD 所使用的实际算法.

1. *QR 迭代及其变形*. 这个算法被适当地执行时[104]，它是求双对角阵所有奇异值最快速的算法. 而且如5.2.1节中所讨论的那样，它求出所有的奇异值达到高的相对精度. 这意味着所有奇异值的全部数位都是正确的，连最微小的奇异值都是这样. 相反，对称三对角 QR 迭代可能计算根本没有相对精度的微小的特征值. QR 迭代的一个不同的变形[80]也用于计算奇异向量：利用零位移的 QR 迭代计算最小的奇异向量，这个变形计算几乎精确的奇异值，还得到如5.2.1节中所述的精确的奇异向量. 但是这仅仅对维数达到 $n=25$ 左右为止的小矩阵是最快速的算法. 这个程序可利用 LAPACK 中子程序 sbdsqr.

2. *分而治之*. 这目前是求维数大于 $n=25$ 的矩阵全部奇异值和奇异向量最快的方法.（在 LAPACK 中执行过程为 sbdsdc，对小矩阵默认为 sbdsqr.）然而，分而治之不保证计算微小的奇异值达到高的相对精度，但它仅保证如对称特征问题中那样相同的误差界：在奇异值 σ_j 中的误差至多不过是 $O(\varepsilon)\sigma_1$ 而不是 $O(\varepsilon)\sigma_j$. 对大多数应用来说这是足够精确了.

[241] 3. *对分法和逆迭代*. 只要求区间中的奇异值可以对引理5.5第1部分的 T_{PS} 应用对分法和逆迭代. 此算法保证求奇异值达到高的相对精度，虽然如5.3.4节中所述奇异向量偶尔可能遭受正交性损失.

4. *雅可比方法*. 可以隐式地(implicitly)对 GG^T 和 G^TG 应用5.3.5节的雅可比法计算稠密阵 G 的 SVD，即不直接构成任何一个而可能丢失稳定性. 对某类 G，即对有利于应用5.2.1节相对扰动理论的那些矩阵如5.2.1节中所述，可以证明雅可比法计算奇异值和奇异向量达到高的相对精度.

下面几节更详尽地描述上面的某些算法，著名的 QR 迭代及其变形 dqds 在5.4.1节；dqds 和对分法的高精度的证明在5.4.2节；雅可比法在5.4.3节. 略去分而治之是因为它总体上相似于5.4.3节中讨论的算法，有关细节建议读者查阅[130].

5.4.1　双对角 SVD 的 QR 迭代及其变形

设计 SVD 的 QR 迭代的变形使其尽可能地有效和精确的历史十分悠久，[200]是一篇优秀的综述文献. 在 LAPACK 程序 sbdsqr 中的算法原来是以[80]为基础，后来当只要求奇异值时利用[104]中的算法作了更新. 由于历史上的原因[1]，后面这个的算法称为 dqds，它是雅致的、快速的和精确的，所以我们将介绍它.

为导出 dqds，用一个发表日期先于 QR 迭代的称为 LR 迭代的算法开始，范围限

---

1. dqds 是“带位移的微分商－差算法”（“differential quotient-difference algorithm with shifts”）的简称[209].

定对称正定矩阵.

> **算法 5.9** LR 迭代：设 $\boldsymbol{T}_0$ 是任意的对称正定矩阵. 下列算法产生一系列相似的对称正定矩阵 $\boldsymbol{T}_i$：
>
> $i=0$
>
> repeat
>
> 选择比 $\boldsymbol{T}_i$ 最小特征值更小的位移 τ_i^2.
>
> 计算楚列斯基分解 $\boldsymbol{T}_i-\tau_i^2\boldsymbol{I}=\boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{B}_i$
>
> （$\boldsymbol{B}_i$ 是具有正对角元的上三角阵）
>
> $\boldsymbol{T}_{i+1}=\boldsymbol{B}_i\boldsymbol{B}_i^{\mathrm{T}}+\tau_i^2\boldsymbol{I}$
>
> $i=i+1$
>
> until 收敛

242

LR 迭代在结构上非常类似于 QR 迭代：计算一个因子分解，反序相乘因子得到下一个迭代 $\boldsymbol{T}_{i+1}$. 容易看出 $\boldsymbol{T}_{i+1}$ 和 $\boldsymbol{T}_i$ 相似：$\boldsymbol{T}_{i+1}=\boldsymbol{B}_i\boldsymbol{B}_i^{\mathrm{T}}+\tau_i^2\boldsymbol{I}=\boldsymbol{B}_i^{-\mathrm{T}}\boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{B}_i\boldsymbol{B}_i^{\mathrm{T}}+\tau_i^2\boldsymbol{B}_i^{-\mathrm{T}}\boldsymbol{B}_i^{\mathrm{T}}=\boldsymbol{B}_i^{-\mathrm{T}}\boldsymbol{T}_i\boldsymbol{B}_i^{\mathrm{T}}$.

事实上，当位移 $\tau_i^2=0$ 时，可以证明 LR 迭代的两步产生像 QR 迭代一步一样相同的 $\boldsymbol{T}_2$.

引理 5.6 设 T_2 是利用 $\tau_i^2=0$ 通过算法 5.9 的两步产生的矩阵，而 T' 是 QR 迭代的一步产生的矩阵（$\boldsymbol{QR}=\boldsymbol{T}_0,\boldsymbol{T}'=\boldsymbol{RQ}$）. 则 $\boldsymbol{T}_2=\boldsymbol{T}'$.

证明 因为 $\boldsymbol{T}_0$ 对称，所以可用两种方式分解 $\boldsymbol{T}_0^2$；第一种分解是 $\boldsymbol{T}_0^2=\boldsymbol{T}_0^{\mathrm{T}}\boldsymbol{T}_0=(\boldsymbol{QR})^{\mathrm{T}}\boldsymbol{QR}=\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}$. 不失一般性. 假设 $\boldsymbol{R}_{ii}>0$. 这是把 $\boldsymbol{T}_0^2$ 分解成下三角阵 $\boldsymbol{R}^{\mathrm{T}}$ 乘它的转置的一个分解；因为楚列斯基分解是唯一的，所以事实上这个分解必定是楚列斯基分解. 第二种分解是 $\boldsymbol{T}_0^2=\boldsymbol{B}_0^{\mathrm{T}}\boldsymbol{B}_0\boldsymbol{B}_0^{\mathrm{T}}\boldsymbol{B}_0$. 现在由算法 5.9，$\boldsymbol{T}_1=\boldsymbol{B}_0\boldsymbol{B}_0^{\mathrm{T}}=\boldsymbol{B}_1^{\mathrm{T}}\boldsymbol{B}_1$，故可改写 $\boldsymbol{T}_0^2=\boldsymbol{B}_0^{\mathrm{T}}\boldsymbol{B}_0\boldsymbol{B}_0^{T}\boldsymbol{B}_0=\boldsymbol{B}_0^{\mathrm{T}}(\boldsymbol{B}_1^{\mathrm{T}}\boldsymbol{B}_1)\boldsymbol{B}_0=(\boldsymbol{B}_1\boldsymbol{B}_0)^{\mathrm{T}}(\boldsymbol{B}_1\boldsymbol{B}_0)$，这也是把 $\boldsymbol{T}_0^2$ 分解成下三角阵 $(\boldsymbol{B}_1\boldsymbol{B}_0)^{\mathrm{T}}$ 乘它的转置的一个分解，故这个分解又必定是楚列斯基分解. 由楚列斯基分解的唯一性，我们断定 $\boldsymbol{R}=\boldsymbol{B}_1\boldsymbol{B}_0$，从而把 LR 迭代的两步与 QR 迭代的一步联系起来. 利用这个关系如下：$\boldsymbol{T}_0=\boldsymbol{QR}$ 可推出

$$
\begin{aligned}
\boldsymbol{T}'&=\boldsymbol{RQ}=\boldsymbol{RQ}(\boldsymbol{RR}^{-1})=\boldsymbol{R}(\boldsymbol{QR})\boldsymbol{R}^{-1}=\boldsymbol{RT}_0\boldsymbol{R}^{-1} && \text{因为 } \boldsymbol{T}_0=\boldsymbol{QR}\\
&=(\boldsymbol{B}_1\boldsymbol{B}_0)(\boldsymbol{B}_0^{\mathrm{T}}\boldsymbol{B}_0)(\boldsymbol{B}_1\boldsymbol{B}_0)^{-1} && \text{因为 } \boldsymbol{R}=\boldsymbol{B}_1\boldsymbol{B}_0\text{，而 }\boldsymbol{T}_0=\boldsymbol{B}_0^{\mathrm{T}}\boldsymbol{B}_0\\
&=\boldsymbol{B}_1\boldsymbol{B}_0\boldsymbol{B}_0^{\mathrm{T}}\boldsymbol{B}_0\boldsymbol{B}_0^{-1}\boldsymbol{B}_1^{-1}=\boldsymbol{B}_1(\boldsymbol{B}_0\boldsymbol{B}_0^{\mathrm{T}})\boldsymbol{B}_1^{-1}\\
&=\boldsymbol{B}_1(\boldsymbol{B}_1^{\mathrm{T}}\boldsymbol{B}_1)\boldsymbol{B}_1^{-1} && \text{因为 } \boldsymbol{B}_0\boldsymbol{B}_0^{\mathrm{T}}=\boldsymbol{T}_1=\boldsymbol{B}_1^{\mathrm{T}}\boldsymbol{B}_1\\
&=\boldsymbol{B}_1\boldsymbol{B}_1^{\mathrm{T}}\\
&=\boldsymbol{T}_2.
\end{aligned}
$$

证毕. □

算法5.9和引理5.6都不依赖于$\boldsymbol{T}_0$是三对角阵，而仅仅是对称$\boldsymbol{T}_0$正定的. 利用引理5.6中LR迭代和QR迭代之间的关系，可以证明把QR迭代的收敛性分析的主要部分转到LR迭代上；在此，我们对其不加以研究.

最终的算法dqds在数学上等价于LR迭代，但它并不如算法5.9中所描述的那样实施，因为这将直接构成$\boldsymbol{T}_{i+1}=\boldsymbol{B}_i\boldsymbol{B}_i^{\mathrm{T}}+\tau_i^2\boldsymbol{I}$，这在5.4中已指出可能数值不稳定. 作为替代，将从$\boldsymbol{B}_i$直接地构成$\boldsymbol{B}_{i+1}$而无需总是构成中间矩阵$\boldsymbol{T}_{i+1}$.

为简化记号，设$\boldsymbol{B}_i$有对角元$a_1,\cdots,a_n$和上对角元$b_1,\cdots,b_{n-1}$，而$\boldsymbol{B}_{i+1}$有对角元$\hat{a}_1,\cdots,\hat{a}_n$和上对角元$\hat{b}_1,\cdots,\hat{b}_{n-1}$. 使用约定$b_0=\hat{b}_0=b_n=\hat{b}_n=0$. 利用

[243]

$$\boldsymbol{B}_{i+1}^{\mathrm{T}}\boldsymbol{B}_{i+1}+\tau_{i+1}^2\boldsymbol{I}=\boldsymbol{T}_{i+1}=\boldsymbol{B}_i\boldsymbol{B}_i^{\mathrm{T}}+\tau_i^2\boldsymbol{I} \tag{5.20}$$

展示$\boldsymbol{B}_i$和$\boldsymbol{B}_{i+1}$之间的联系. 对$j<n$使(5.20)式左边和右边的j，j元素相等，得到

$$\hat{a}_j^2+\hat{b}_{j-1}^2+\tau_{i+1}^2=a_j^2+b_j^2+\tau_i^2 \text{ 或 } \hat{a}_j^2=a_j^2+b_j^2-\hat{b}_{j-1}^2-\delta, \tag{5.21}$$

其中$\delta=\tau_{i+1}^2-\tau_i^2$. 因为从下面必须选择$\tau_i^2$接近$T$的最小特征值(保持$T_i$正定以及算法有意义)，$\delta\geqslant 0$. 使(5.20)式左边和右边的$j$，$j+1$元素相等，得到

$$\hat{a}_j^2\hat{b}_j^2=a_{j+1}^2b_j^2 \text{ 或 } \hat{b}_j^2=a_{j+1}^2b_j^2/\hat{a}_j^2. \tag{5.22}$$

合并(5.21)式和(5.22)式得到尚未最终确定的算法：

for $j=1$ to $n-1$
  $\hat{a}_j^2=a_j^2+b_j^2-\hat{b}_{j-1}^2-\delta$
  $\hat{b}_j^2=b_j^2\cdot(a_{j+1}^2/\hat{a}_j^2)$
end for
$\hat{a}_n^2=a_n^2-\hat{b}_{n-1}^2-\delta$

算法的这个形式在内循环中只有5个浮点运算，这是十分便宜的. 它把$\boldsymbol{B}_i$的元素的平方直接映射到$\boldsymbol{B}_{i+1}$的元素的平方. 在算法真正终止之前没有必要取平方根. 实际上，在现代计算机上，平方根与除法比加法、减法或乘法所需时间长10到30倍，故应该尽可能避免它们. 为强调我们正在计算元素的平方，把变量改为$q_j\equiv a_j^2$和$e_j\equiv b_j^2$，得到倒数第二的算法qds(名称还是与我们无关系的历史原因[209]).

算法5.10 qds算法的单步
for $j=1$ to $n-1$
  $\hat{q}_j=q_j+e_j-\hat{e}_{j-1}-\delta$
  $\hat{e}_j=e_j\cdot(q_{j+1}/\hat{q}_j)$
end for
$\hat{q}_n=q_n-\hat{e}_{n-1}-\delta$

最终的算法dqds将与qds的工作量差不多相同但(正如5.4.2节中将指出的那样)更为准确. 从算法5.10的第一行取子表达式$q_j-\hat{e}_{j-1}-\delta$并改写它为

244

$$
\begin{aligned}
d_j &\equiv q_j - \hat{e}_{j-1} - \delta \\
&= q_j - \frac{q_j e_{j-1}}{\hat{q}_{j-1}} - \delta \quad \text{从}(5.22) \\
&= q_j \cdot \left[\frac{\hat{q}_{j-1} - e_{j-1}}{\hat{q}_{j-1}}\right] - \delta \\
&= q_j \cdot \left[\frac{q_{j-1} - \hat{e}_{j-2} - \delta}{\hat{q}_{j-1}}\right] - \delta \quad \text{从}(5.21) \\
&= \frac{q_j}{\hat{q}_{j-1}} \cdot d_{j-1} - \delta.
\end{aligned}
$$

由此改写算法 5.10 的内循环为

$$
\begin{aligned}
\hat{q}_j &= d_j + e_j \\
\hat{e}_j &= e_j \cdot (q_{j+1}/\hat{q}_j) \\
d_{j+1} &= d_j \cdot (q_{j+1}/\hat{q}_j) - \delta
\end{aligned}
$$

最后，注意 d_{j+1} 可覆盖 d_j，并且为得到最后的 dqds 算法，$t = q_{j+1}/\hat{q}_j$ 只需计算一次.

算法 5.11 单步 dqds 算法

> $d = q_1 - \delta$
> for $j = 1$ to $n-1$
> $\hat{q}_j = d + e_j$
> $t = (q_{j+1}/\hat{q}_j)$
> $\hat{e}_j = e_j \cdot t$
> $d = d \cdot t - \delta$
> end for
> $\hat{q}_n = d$

dqds 算法在它的内循环中有如 qds 那样相同的浮点运算次数，但以一个减法换一个乘法．正如下节中所述，这个修正出色地保证了高的相对精度的顺利实现.

还有两个重要的问题没有讨论：选择位移 $\delta = \tau_{i+1}^2 - \tau_i^2$ 以及检测收敛性．这些在 [104] 中详尽地加以讨论.

5.4.2 计算双对角 SVD 达到高的相对精度

本节依赖于 5.2.1 节，首次阅读可以跳过.

计算双对角矩阵 $\boldsymbol{B}$ 的 SVD 达到高的相对精度（如 5.2.1 节中定义的）的能力依赖于下面的定理 5.13，它指出 $\boldsymbol{B}$ 的元素中小的相对改变只引起奇异值中小的相对改变.

[245] **引理5.7** 设B是具有对角元$a_1,\cdots,a_n$和上对角元$b_1,\cdots,b_{n-1}$的双对角阵. 设$\hat{B}$是具有对角元$\hat{a}_i=a_i\chi_i$和上对角元$\hat{b}_i=b_i\zeta_i$的另一个双对角阵. 则$\hat{B}=D_1BD_2$，其中

$$D_1=\operatorname{diag}\left(\chi_1,\frac{\chi_2\chi_1}{\zeta_1},\frac{\chi_3\chi_2\chi_1}{\zeta_2\zeta_1},\cdots,\frac{\chi_n\cdots\chi_1}{\zeta_{n-1}\cdots\zeta_1}\right),$$

$$D_2=\operatorname{diag}\left(1,\frac{\zeta_1}{\chi_1},\frac{\zeta_2\zeta_1}{\chi_2\chi_1},\cdots,\frac{\zeta_{n-1}\cdots\zeta_1}{\chi_{n-1}\cdots\chi_1}\right).$$

这个定理的证明是一个简单的计算(见问题5.20). 现在可应用推论5.2推出下列定理.

定理5.13 设B为$\hat{B}$如引理5.7中那样定义. 假如存在一个$\tau\geq1$使得$\tau^{-1}\leq\chi_i\leq\tau$且$\tau^{-1}\leq\zeta_i\leq\tau$. 换言之$\varepsilon\equiv\tau-1$是$B$的每个元素和相应的$\hat{B}$的元素之间相对差的一个界. 设$\sigma_n\leq\cdots\leq\sigma_1$是$B$的奇异值而$\hat{\sigma}_n\leq\cdots\leq\hat{\sigma}_1$是$\hat{B}$的奇异值. 则$|\hat{\sigma}_i-\sigma_i|\leq\sigma_i(\tau^{4n-2}-1)$. 若$\sigma_i\neq0$且$\tau-1=\varepsilon\ll1$，则可记

$$\frac{|\hat{\sigma}_i-\sigma_i|}{\sigma_i}\leq\tau^{4n-2}-1=(4n-2)\varepsilon+O(\varepsilon^2).$$

因而奇异值中的相对改变$|\hat{\sigma}_i-\sigma_i|/\sigma_i$是以矩阵元素中的相对改变$\varepsilon$的$4n-2$倍为界. 用稍多一点操作，因子$4n-2$可改进为$2n-1$(见问题5.21). 也可指出能十分精确地确定奇异向量，如5.2.1节中定义的那样，与相对间隙的倒数成正比.

我们将证明对分法(算法5.4应用于引理5.5中的T_{ps})和dqds(算法5.11)两者都能用于求双对角阵的奇异值达到高的相对精度. 首先考虑对分法. 记得对称三对角阵T_{ps}的特征值是B的奇异值及它们的负数的平方. 引理5.3可推得利用(5.17)式计算的$T_{ps}-\lambda I$的惯性是某个$\hat{B}$的惯性，其中$\hat{B}$和B相应的元素之相对差至多是2.5ε左右. 所以由定理5.13计算的奇异值($\hat{B}$的奇异值)和真正的奇异值之间的相对差至多是$(10n-5)\varepsilon$左右.

现在考虑算法5.11. 我们将利用定理5.13证明B的奇异值(输入到算法5.11)和$\hat{B}$的奇异值(从算法5.11中输出)符合高的相对精度. 这个事实可推出在dqds的许多步之后，当$\hat{B}$几乎是对角线上具有它的奇异值的对角阵时，这些奇异值匹配原输入矩阵的奇异值达到高的相对精度.

[246] 便于理解的最简单情况是位移$\delta=0$时. 此时，在dqds中仅有的运算是正数的加法，乘法和除法；不发生对消. 粗略地讲，这些基本运算构成的表达式序列保证计算每个输出达到高的相对精度. 所以计算$\hat{B}$达到高的相对精度，并且由定理5.13知B和$\hat{B}$的奇异值符合高的相对精度. $\delta>0$的一般情况是比较棘手的[104].

定理5.14 以浮点算术运算对B应用算法5.11的单步得到$\hat{B}$等价于下列运算序列：

1. 在B的每个元素中作一个小的相对改变(至多1.5ε)得到$\tilde{B}$.

2. 以精确算术运算对 $\tilde{\boldsymbol{B}}$ 应用算法5.11 的单步，得到 $\check{\boldsymbol{B}}$.

3. 在 $\check{\boldsymbol{B}}$ 的每个元素中作一个小的相对改变(至多 ε)，得到 $\hat{\boldsymbol{B}}$.

上面第1步和第3步在双对角阵的奇异值中仅仅作小的相对改变，故由定理5.13知 $\boldsymbol{B}$ 和 $\hat{\boldsymbol{B}}$ 的奇异值符合高的相对精度.

证明 记算法5.11的内循环如下，对变量 d 和 t 引入下标以保存它们在不同的迭代中的踪迹并包括关于舍入误差的带下标的 $1+\varepsilon$ 项：

$$\hat{q}_j=(d_j+e_j)(1+\varepsilon_{j,+})$$
$$t_j=(q_{j+1}/\hat{q}_j)(1+\varepsilon_{j,/})$$
$$\hat{e}_j=e_j\cdot t_j(1+\varepsilon_{j,*1})$$
$$d_{j+1}=(d_j\cdot t_j(1+\varepsilon_{j,*2})-\delta)(1+\varepsilon_{j,-})$$

把第一行代入到第二行得到

$$t_j=\frac{q_{j+1}}{d_j+e_j}\cdot\frac{1+\varepsilon_{j,/}}{1+\varepsilon_{j,+}}.$$

把这个 t_j 的表达式代入到算法的最后一行并用 $1+\varepsilon_{j,-}$ 相除得到

$$\frac{d_{j+1}}{1+\varepsilon_{j,-}}=\frac{d_jq_{j+1}}{d_j+e_j}\cdot\frac{(1+\varepsilon_{j,/})(1+\varepsilon_{j,*2})}{1+\varepsilon_{j,+}}-\delta. \tag{5.23}$$

这告诉我们如何确定 $\tilde{B}$：设

$$\tilde{d}_{j+1}=\frac{d_{j+1}}{1+\varepsilon_{j,-}},$$
$$\tilde{e}_j=\frac{e_j}{1+\varepsilon_{j-1,-}}, \tag{5.24}$$
$$\tilde{q}_{j+1}=q_{j+1}\frac{(1+\varepsilon_{j,/})(1+\varepsilon_{j,*2})}{1+\varepsilon_{j,+}},$$

247

故(5.23)变成

$$\tilde{d}_{j+1}=\frac{\tilde{d}_j\,\tilde{q}_{j+1}}{\tilde{d}_j+\tilde{e}_j}-\delta.$$

注意从(5.24)知，$\tilde{\boldsymbol{B}}$ 不同于 $\boldsymbol{B}$ 是在每个元素中至多为 1.5ε 的相对改变(从 $\tilde{q}_{j+1}=\tilde{a}^2_{j+1,j+1}$ 中的三个 $1+\varepsilon$ 因子).

现在可用下式定义 $\check{\boldsymbol{B}}$ 中的 $\check{q}_j$ 和 $\check{e}_j$：

$$\check{q}_j=\tilde{d}_j+\tilde{e}_j$$
$$\tilde{t}_j=(\tilde{q}_{j+1}/\check{q}_j),$$
$$\check{e}_j=\tilde{e}_j\cdot\tilde{t}_j,$$
$$\tilde{d}_{j+1}=\tilde{d}_j.\ \tilde{t}_j-\delta.$$

这是dqds算法精确地应用于 $\boldsymbol{B}$ 的一步，得到 $\check{\boldsymbol{B}}$. 为最后证明 $\check{\boldsymbol{B}}$ 不同于 $\hat{\boldsymbol{B}}$ 是在每个元

素中至多为 ε 的相对改变，注意

$$\begin{aligned}\check{q}_j &= \tilde{d}_j + \tilde{e}_j \\ &= \frac{d_j}{1+\varepsilon_{j-1,-}} + \frac{e_j}{1+\varepsilon_{j-1,-}} \\ &= (d_j+e_j)(1+\varepsilon_{j,+}) \cdot \frac{1}{(1+\varepsilon_{j,+})(1+\varepsilon_{j-1,-})} \\ &= \hat{q}_j \cdot \left[\frac{1}{(1+\varepsilon_{j,+})(1+\varepsilon_{j-1,-})}\right]\end{aligned}$$

以及

$$\begin{aligned}\check{e}_j &= \tilde{e}_j \cdot \tilde{t}_j \\ &= \frac{e_j}{1+\varepsilon_{j-1,-}} \cdot \frac{\tilde{q}_{j+1}}{\check{q}_j} \\ &= \frac{e_j}{1+\varepsilon_{j-1,-}} \cdot t_j(1+\varepsilon_{j,*2})(1+\varepsilon_{j-1,-}) \\ &= e_j t_j (1+\varepsilon_{j,*1}) \frac{1+\varepsilon_{j,*2}}{1+\varepsilon_{j,*1}} \\ &= \hat{e}_j \cdot \left[\frac{1+\varepsilon_{j,*2}}{1+\varepsilon_{j,*1}}\right].\end{aligned}$$

□

5.4.3　SVD 的雅可比法

在 5.3.5 节中讨论了求稠密阵 $\boldsymbol{A}$ 的特征值和特征向量的雅可比法，并说明它是 [248] 这个问题可以利用的最慢的方法．在本节中将通过对对称阵 $\boldsymbol{A}=\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$ 隐式地应用 5.3.5 节的算法 5.8 来指出如何应用雅可比法求稠密矩阵 $\boldsymbol{G}$ 的 SVD．这蕴含着此方法的收敛性质几乎与算法 5.8 的那些结论相同，特别是雅可比方法也是 SVD 可利用的最慢的方法．

然而由于对某些类型的矩阵 $\boldsymbol{G}$，它比其他已经讨论过的算法能够更加精确地计算奇异值和奇异向量．因此，雅可比方法还是有吸引力的．对这些 $\boldsymbol{G}$，雅可比方法如 5.2.1 节中所述计算奇异值和奇异向量达到高的相对精度．

在描述关于 $\boldsymbol{G}$ 的 SVD 的隐式雅可比方法之后，将证明当 $\boldsymbol{G}$ 可写成形式 $\boldsymbol{G}=\boldsymbol{D}\boldsymbol{X}$ 时，它计算 SVD 达到高的相对精度，其中 $\boldsymbol{D}$ 为对角阵并且 $\boldsymbol{X}$ 是良态的．（这意味着 $\boldsymbol{G}$ 病态当且仅当 $\boldsymbol{D}$ 同时有大的和小的对角元）．更一般地，在 $\boldsymbol{X}$ 比 $\boldsymbol{G}$ 是有较好的性态时对我们有益．我们将用一个矩阵来说明这点，其中任何涉及化双对角形式的算法必定丢失除最大奇异值以外所有奇异值的一切有效数字，而雅可比方法计算所有的奇异值到全机器精度．然后概述其他类型的矩阵 $\boldsymbol{G}$，对那些 $\boldsymbol{G}$ 用雅可比方法也比利用双对角化的方法更为准确．

注意若 G 为双对角阵，则在 5.4.2 节中指出可利用对分法或 dqds 算法(5.4.1 节)计算 SVD 达到高的相对精度. 问题是把一个矩阵从稠密形式化到双对角形式可能引入足够大的破坏高的相对精度的误差，正如将指出的例子那样. 因为雅可比方法对原矩阵而不是首先把它化为双对角形式操作，所以在更多情况下它可以达到高的相对精度.

隐式雅可比方法在数学上等价于对 $A=G^TG$ 应用算法 5.8. 换言之，在每一步计算一个雅可比旋转并隐式地更新 G^TG 至 J^TG^TGJ，其中选择 J 使 G^TG 的两个非对角元在 J^TG^TGJ 中置为零. 但不是显式地计算 G^TG 和 J^TG^TGJ 而只计算 GJ. 由于这个原因，称这个算法为单边雅可比旋转(one-sided Jacobi rotation).

算法 5.12 对 G 关于坐标 j, k 计算和应用单边雅可比旋转：

proc One-Sided-Jacobi-Rotation(G,j,k)
计算 $a_{jj}=(G^TG)_{jj}, a_{jk}=(G^TG)_{jk}$, 和 $a_{kk}=(G^TG)_{kk}$
if $|a_{jk}|$ 不是太小
  $\tau=(a_{jj}-a_{kk})/(2\cdot a_{jk})$
  $t=\mathrm{sign}(\tau)/(|\tau|+\sqrt{1+\tau^2})$
  $c=1/\sqrt{1+t^2}$
  $s=c\cdot t$
  $G=G\cdot R(j,k,\theta)$ …其中 $c=\cos\theta$ 和 $s=\sin\theta$
  if 要求右奇异向量
    $J=J\cdot R(j,k,\theta)$
  end if
end if

249

注意 $A=G^TG$ 的 jj, jk 和 kk 元素是由过程单边雅可比旋转计算的，之后它以过程雅可比旋转(算法 5.5)同样的方法计算雅可比旋转 $R(j,k,\theta)$.

算法 5.13 单边雅可比：假定 G 是 $n\times n$ 阶矩阵. 输出是奇异值 σ_i，左奇异向量矩阵 U 和右奇异向量矩阵 V，所以 $G=U\Sigma V^T$，其中 $\Sigma=\mathrm{diag}(\sigma_i)$.

repeat
  for $j=1$ to $n-1$
    for $k=j+1$ to n
      调用 One-Sided-Jacobi-Rotation(G,j,k)
    end for
  end for
until G^TG 足够对角化

（续）

设 $\sigma_i = \|G(:,i)\|_2$（G 的第 i 列之 2-范数） 设 $U = [u_1, \cdots, u_n]$，其中 $u_i = G(:,i)/\sigma_i$ 设 $V = J$，雅可比旋转累计的乘积.

问题5.22要求证明用单边雅可比计算的Σ，U和V确实构成G的SVD.

下列定理证明：倘若可记$G = DX$，其中D为对角阵且X良态，尽管舍入，单边雅可比可计算SVD达到高的相对精度.

定理5.15　设$G = DX$是$n \times n$阶矩阵，其中D为非奇异对角阵，且X非奇异. 设$\hat{G}$是执行按浮点算术运算的One-Sided-Jacobi-Rotation(G,j,k) m次以后的矩阵. 设$\sigma_1 \geq \cdots \geq \sigma_n$是$G$的奇异值，$\hat{\sigma}_1 \geq \cdots \geq \hat{\sigma}_n$是$\hat{G}$的奇异值. 则

$$\frac{|\sigma_i - \hat{\sigma}_i|}{\sigma_i} \leq O(m\varepsilon)\kappa(X), \tag{5.25}$$

其中$\kappa(X) = \|X\| \cdot \|X^{-1}\|$是$X$的条件数. 换言之，当$X$的条件数小时奇异值中的相对误差也小.

证明　首先考虑$m = 1$；即只应用单个雅可比旋转. 并且后面推广至较大的m.

考察One-Sided-Jacobi-Rotation(G,j,k)，我们看到$\hat{G} = \mathrm{fl}(G \cdot \tilde{R})$，其中$\tilde{R}$为浮点吉文斯旋转. 由构造$\tilde{R}$和某个精确的吉文斯旋转$R$不同，按范数相差$O(\varepsilon)$.（$\tilde{R}$和“真正的”雅可比旋转相差$O(\varepsilon)$是不重要的和不一定成立的，One-Sided-Jacobi-Rotation(G,j,k)应当以精确的算术运算计算，它与某个旋转相差$O(\varepsilon)$是必要的. 容易验证，这只需要$c^2 + s^2 = 1 + O(\varepsilon)$.）

我们的目标是证明$\hat{G} = GR(I+E)$，对某个按范数是小的E：$\|E\|_2 = O(\varepsilon)\kappa(X)$. 若$E$为零，则$\hat{G}$和$GR$将有相同的奇异值，因为$R$是精确地正交的. 当$E$按范数小于1时，可利用推论5.2界定奇异值中的相对差为

$$\frac{|\sigma_i - \hat{\sigma}_i|}{\sigma_i} \leq \|(I+E)(I+E)^{\mathrm{T}} - I\|_2 = \|E + E^{\mathrm{T}} + EE^{\mathrm{T}}\|_2 \leq 3\|E\|_2 = O(\varepsilon)\kappa(X). \tag{5.26}$$

现在构造E. 因为$\tilde{R}$右乘G，所以$\hat{G}$的每行只与G相应的行有关；按Matlab中的记号把这个记为$\hat{G}(i,:) = \mathrm{fl}(G(i,:) \cdot \hat{R})$. 设$F = \hat{G} - GR$. 则由引理3.1和$G = DX$的事实，

$$\|F(i,:)\|_2 = \|\hat{G}(i,:) - G(i,:)R\|_2 = O(\varepsilon)\|G(i,:)\|_2 = O(\varepsilon)\|d_{ii}X(i,:)\|_2$$

因而$\|d_{ii}^{-1}F(i,:)\|_2 = O(\varepsilon)\|X(i,:)\|_2$或$\|D^{-1}F\|_2 = O(\varepsilon)\|X\|_2$. 因为$R^{-1} = R^{\mathrm{T}}$和$G^{-1} = (DX)^{-1} = X^{-1}D^{-1}$，所以

$$\hat{G} = GR + F = GR(I + R^{\mathrm{T}}G^{-1}F) = GR(I + R^{\mathrm{T}}X^{-1}D^{-1}F) \equiv GR(I+E)$$

其中

$$\|E\|_2 \leqslant \|R^{\mathrm{T}}\|_2 \|X^{-1}\|_2 \|D^{-1}F\|_2 = O(\varepsilon)\|X\|_2 \|X^{-1}\|_2 = O(\varepsilon)\kappa(X).$$

把这个结果推广到 $m>1$ 次旋转，注意，按精确的算术运算应该有 $\hat{G}=GR=DXR=D\hat{X}$ 和 $\kappa(\hat{X})=\kappa(X)$，所以 m 步中的每一步应该应用界(5.26)，得到界(5.25)．因为舍入，所以在每一步上 $\kappa(\hat{X})$ 能差不多以 $\kappa(I+E)\leqslant(1+O(\varepsilon)\kappa(X))$ 增长，这是一个非常接近于1的因子，我们把它并入到 $O(m\varepsilon)$ 项中． □

为了使算法完整，需要注意停止准则，即如何实施算法 5.12 One-Sided-Jacobi-Rotation 中的语句"若 $|a_{jk}|$ 不是太小"．适当的准则

$$|a_{jk}| \geqslant \varepsilon \sqrt{a_{jj}a_{kk}}$$

在问题 5.24 中进一步讨论．

251

例 5.9 考虑一个极端的例子 $G=DX$，其中雅可比法计算全部奇异值达到全机器精度；依赖于双对角化只计算最大的一个奇异值 $\sqrt{3}$ 达到全机器精度的任何方法；以及所有其他根本不准确的方法（虽然从向后稳定算法来看不出所料它计算奇异值仍然具有误差 $\pm O(\varepsilon)\cdot\sqrt{3}$）．在本例中 $\varepsilon\approx 2^{-53}\approx 10^{-16}$．（*IEEE* 双精度），$\eta=10^{-20}$（接近于 $\eta<\varepsilon$ 的任何值可用）．定义

$$G \equiv \begin{bmatrix} \eta & 1 & 1 & 1 \\ \eta & \eta & 0 & 0 \\ \eta & 0 & \eta & 0 \\ \eta & 0 & 0 & \eta \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & \eta & & \\ & & \eta & \\ & & & \eta \end{bmatrix} \cdot \begin{bmatrix} \eta & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \equiv D \cdot X.$$

G 的奇异值是 $\sqrt{3},\sqrt{3}\cdot\eta,\eta$ 和 η，至少 16 位数字．为观察把 G 化为双对角形式精度将如何丢失，我们只考虑 4.4.7 节中算法的第 1 步；第 1 步以后用豪斯霍尔德变换左乘使 $G(2:4,1)$ 化为零，按精确的算术运算 G 应该是

$$\begin{bmatrix} -2\eta & -0.5-\frac{\eta}{2} & -0.5-\frac{\eta}{2} & -0.5-\frac{\eta}{2} \\ 0 & -0.5+\frac{5\eta}{6} & -0.5-\frac{\eta}{6} & -0.5-\frac{\eta}{6} \\ 0 & -0.5-\frac{\eta}{6} & -0.5+\frac{5\eta}{6} & -0.5-\frac{\eta}{6} \\ 0 & -0.5-\frac{\eta}{6} & -0.5-\frac{\eta}{6} & -0.5+\frac{5\eta}{6} \end{bmatrix}$$

但是因为 η 是如此之小，所以它舍入成

$$G_1 = \begin{bmatrix} -2\eta & -0.5 & -0.5 & -0.5 \\ 0 & -0.5 & -0.5 & -0.5 \\ 0 & -0.5 & -0.5 & -0.5 \\ 0 & -0.5 & -0.5 & -0.5 \end{bmatrix}.$$

注意关于 η 的所有信息从 G_1 的最后三列中"丢失"．因为 G_1 的最后三列是恒等的，所以 G_1 刚好是奇异的，而且实际上秩为 2．因此两个最小的奇异值从 η 变化为 0，相

对精度完全丢失．若不产生进一步舍入误差的话，将把 G_1 化为双对角形式

$$B=\begin{bmatrix} -2\eta & \sqrt{0.75} & & \\ & 1.5 & 0 & \\ & & 0 & 0 \\ & & & 0 \end{bmatrix}$$

具有奇异值 $\sqrt{3}$，$\sqrt{3}\eta$，0 和 0，其中最大的两个是 G 的准确的奇异值．但是，当继续进行把 G_1 化为双对角形式的算法时，舍入引入 $O(\varepsilon)$ 的非零数量到 B 的非零元中，使所有三个小的奇异值不准确．两个最小的非零计算奇异值是舍入的意外事故且与 ε 成比例．

252

对这个矩阵单边雅可比法没有困难，在第三次扫描中收敛于 $G=U\Sigma V^T$，其中

$$U=\begin{bmatrix} 0 & -\frac{\eta^2}{\sqrt{2}} & 1 & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{\eta}{3} & \frac{-1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{\eta}{3} & \frac{-1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{\eta}{\sqrt{2}} & \frac{\eta}{3} & \frac{2}{\sqrt{6}} \end{bmatrix},\quad V=\begin{bmatrix} 1 & \frac{\eta}{\sqrt{2}} & \frac{\eta}{\sqrt{3}} & \frac{-\eta}{\sqrt{6}} \\ -\eta & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{6}} \\ 0 & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{6}} \\ 0 & \frac{-\eta^3}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} \end{bmatrix},$$

和 $\Sigma=\mathrm{diag}(\sqrt{3}\eta,\eta,\sqrt{3},\eta)$ 达到机器精度．（雅可比法不自动地整理奇异值；它可作为一个后加工步操作）．◇

下面是其他的一些例子，可以指出雅可比法的变形保证 SVD（或对称特征分解）中高的相对精度，而依赖于双对角化（或三对角化）的方法可能丢失最小奇异值（或特征值）中所有的有效数字．许多其他的例子出现在[75]中．

1. 设 $A=LL^T$ 是对称正定矩阵的楚列斯基分解，则 $L=U\Sigma V^T$ 的 SVD 提供 $A=U\Sigma^2U^T$ 的特征分解．若 $L=DX$，其中 X 是良态而 D 是对角阵，则定理 5.15 告诉我们可以利用雅可比法计算 L 的奇异值 σ_i 达到高的相对精度，并用 $O(\varepsilon)\kappa(X)$ 界定相对误差．但是我们必须说明计算楚列斯基因子 L 中的舍入误差：利用楚列斯基的向后误差界(2.16)（连同定理 5.6 一起）可以界定在楚列斯基分解期间以 $O(\varepsilon)\kappa^2(X)$ 舍入引入到奇异值中的相对误差．因此当 X 良态时，将计算 A 的所有特征值达到高的相对精度（见问题 5.23 和[82,92,183]）．

例 5.10　正如例 5.9 中那样，选择一种极端情况，依赖于最初化 A 为三对角形式的任何算法保证丢失最小的特征值中全部相对精度，而楚列斯基分解随后对楚列斯基因子用单边雅可比法计算全部特征值达到几乎全机器精度．正如在那个例子中那样，设 $\eta=10^{-20}$（任何 $\eta<\varepsilon/120$ 可用），设

$$A=\begin{bmatrix}1 & \sqrt{\eta} & \sqrt{\eta}\\ \sqrt{\eta} & 1 & 10\eta\\ \sqrt{\eta} & 10\eta & 100\eta\end{bmatrix}=\begin{bmatrix}1 & 10^{-10} & 10^{-10}\\ 10^{-10} & 1 & 10^{-19}\\ 10^{-10} & 10^{-19} & 10^{-20}\end{bmatrix}.$$

若精确地把 A 化为三对角形式 T，则

$$T=\begin{bmatrix}1 & \sqrt{2\eta} & \\ \sqrt{2\eta} & 0.5+60\eta & 0.5-50\eta\\ & 0.5-50\eta & 0.5+40\eta\end{bmatrix},$$

253

但是因为 η 是如此之小，所以这个矩阵舍入成

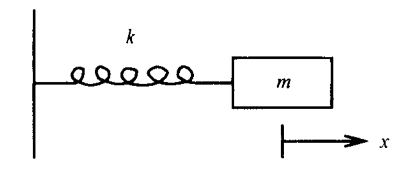
$$\hat{T}=\begin{bmatrix}1 & \sqrt{2\eta} & \\ \sqrt{2\eta} & 0.5 & 0.5\\ & 0.5 & 0.5\end{bmatrix},$$

这个矩阵连正定都不是的，因为右下部 2×2 阶子矩阵恰好是奇异的. 因此 $\hat{T}$ 的最小特征值非正，因而三对角约化丢失最小特征值中全部相对精度. 相反，单边雅可比法没有困难计算 A 的特征值的正确的平方根几乎达到全机器精度，即 $1+\sqrt{\eta}=1+10^{-10}$，$1-\sqrt{\eta}=1-10^{-10}$ 以及 $0.99\eta=0.99\cdot10^{-20}$. ◇

2. 最一般情况是当我们能准确地计算任意的分解 $A=YDX$，其中 X 和 Y 是良态的但其他方面任意，D 是对角阵时懂得如何计算 A 的 SVD 达到高的相对精度. 在最近的例中有 $L=DX$；即 Y 是单位阵. 全主元的高斯消元法是这种分解的另一个来源（Y 为下三角阵而 X 为上三角阵）. 细节见[74]. 这种思想应用于不定对称特征问题，见[228,250]，以及广义对称特征值问题，见[66,92].

5.5 微分方程和特征值问题

从物理学中的守恒律来寻找本节的动力. 再次考虑例 4.1 中引入的并在例 5.1 中再研究的质点 - 弹簧系统. 从一个弹簧和一个质量且无摩擦力的最简单情况开始：



设 x 表示偏离平衡的水平位移. 则牛顿定律 $F=ma$ 变成 $m\ddot{x}(t)+kx(t)=0$. 设 $E(t)=\frac{1}{2}m\dot{x}^2(t)+\frac{1}{2}kx^2(t)=$"动能"+"势能". 力的守恒告诉我们 $\frac{\mathrm{d}}{\mathrm{d}t}E(t)$ 应该为

零. 通过计算$\frac{\mathrm{d}}{\mathrm{d}t}E(t)=m\dot{x}(t)\ddot{x}(t)+kx(t)\dot{x}(t)=\dot{x}(t)(m\ddot{x}(t)+kx(t))=0$可确认这是成立的.

更一般地有“$\boldsymbol{M}\ddot{x}(t)+\boldsymbol{K}x(t)=0$”. 其中$\boldsymbol{M}$是质量矩阵而$K$是刚度矩阵. 能量被定义为$E(t)=\frac{1}{2}\dot{x}^{\mathrm{T}}(t)\boldsymbol{M}\dot{x}(t)+\frac{1}{2}x^{\mathrm{T}}(t)\boldsymbol{K}x(t)$. 通过验证它是守恒的确认这是正确的定义:

254

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}E(t)&=\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\dot{x}^{\mathrm{T}}(t)\boldsymbol{M}\dot{x}(t)+\frac{1}{2}x^{\mathrm{T}}(t)\boldsymbol{K}x(t)\right)\\
&=\frac{1}{2}\left(\ddot{x}^{\mathrm{T}}(t)\boldsymbol{M}\dot{x}(t)+\dot{x}^{\mathrm{T}}(t)\boldsymbol{M}\ddot{x}(t)+\dot{x}^{\mathrm{T}}(t)\boldsymbol{K}x(t)+x^{\mathrm{T}}(t)\boldsymbol{K}\dot{x}(t)\right)\\
&=\dot{x}^{\mathrm{T}}(t)\boldsymbol{M}\ddot{x}(t)+\dot{x}^{\mathrm{T}}(t)\boldsymbol{K}x(t)\\
&=\dot{x}^{\mathrm{T}}(t)\left(\boldsymbol{M}\ddot{x}(t)+\boldsymbol{K}x(t)\right)=0,
\end{aligned}
$$

其中我们使用了$\boldsymbol{M}$和$\boldsymbol{K}$的对称性.

微分方程$\boldsymbol{M}\ddot{x}(t)+\boldsymbol{K}x(t)=0$是线性的. 某些非线性微分方程也使“能量”守恒是一个值得注意的事实.

5.5.1　Toda 格子

为记号简易, 当从上下文知变量是清楚的时候将把$\dot{x}(t)$改记为$\dot{x}$.

Toda 格子(Toda lattice)也是一个质点-弹簧系统. 但是来自弹簧的力是它的伸长的指数衰减函数而不是线性函数:

$$\ddot{x}_i=\mathrm{e}^{-(x_i-x_{i-1})}-\mathrm{e}^{-(x_{i+1}-x_i)}.$$

我们利用边界条件$\mathrm{e}^{-(x_1-x_0)}=0$(即$x_0=-\infty$)和$\mathrm{e}^{-(x_{n+1}-x_n)}=0$(即$x_{n+1}=+\infty$). 更简单地, 这些边界条件意味着在左边和右边不存在墙壁. (见图4-1).

现在把变量改为$b_k=\frac{1}{2}\mathrm{e}^{(x_k-x_{k+1})/2}$和$a_k=-\frac{1}{2}\dot{x}_k$. 这就得到微分方程

$$
\begin{aligned}
\dot{b}_k&=\frac{1}{2}\mathrm{e}^{(x_k-x_{k+1})/2}\cdot\frac{1}{2}(\dot{x}_k-\dot{x}_{k+1})=b_k(a_{k+1}-a_k),\\
\dot{a}_k&=-\frac{1}{2}\ddot{x}_k=2(b_k^2-b_{k-1}^2)
\end{aligned}
\tag{5.27}
$$

和$b_0\equiv0$及$b_n\equiv0$. 现在定义两个三对角阵

$$
\boldsymbol{T}=\begin{bmatrix}a_1&b_1&&\\b_1&\ddots&\ddots&\\&\ddots&\ddots&b_{n-1}\\&&b_{n-1}&a_n\end{bmatrix}\text{和}\boldsymbol{B}=\begin{bmatrix}0&b_1&&\\-b_1&\ddots&\ddots&\\&\ddots&\ddots&b_{n-1}\\&&-b_{n-1}&0\end{bmatrix},
$$

其中 $B = -B^{\mathrm{T}}$. 可以容易进一步证实(5.27)式与 $\frac{\mathrm{d}T}{\mathrm{d}t} = BT - TB$ 是相同的. 这称为 Toda流率(Toda flow).

定理5.16 对一切 t, $T(t)$ 与 $T(0)$ 一样有相同的特征值. 换言之, 特征值如同"能量"一样, 通过微分方程守恒.

255

证明 定义 $\frac{\mathrm{d}}{\mathrm{d}t}U = BU, U(0) = I$. 我们要求对一切 t, $U(t)$ 是正交的. 因为 $U^{\mathrm{T}}U(0) = I$, 所以要证明这点, 只要证明 $\frac{\mathrm{d}}{\mathrm{d}t}U^{\mathrm{T}}U = 0$ 就足够了. 因为 B 反对称, 所以

$$\frac{\mathrm{d}}{\mathrm{d}t}U^{\mathrm{T}}U = \dot{U}^{\mathrm{T}}U + U^{\mathrm{T}}\dot{U} = U^{\mathrm{T}}B^{\mathrm{T}}U + U^{\mathrm{T}}BU = -U^{\mathrm{T}}BU + U^{\mathrm{T}}BU = 0.$$

现在要求 $T(t) = U(t)T(0)U^{\mathrm{T}}(t)$ 满足 Toda 流率 $\frac{\mathrm{d}T}{\mathrm{d}t} = BT - TB$, 意指每个 $T(t)$ 正交相似于 $T(0)$, 故有相同的特征值:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}T(t) &= \dot{U}(t)T(0)U^{\mathrm{T}}(t) + U(t)T(0)\dot{U}^{\mathrm{T}}(t)\\ &= B(t)U(t)T(0)U^{\mathrm{T}}(t) + U(t)T(0)U^{\mathrm{T}}(t)B^{\mathrm{T}}(t)\\ &= B(t)T(t) - T(t)B(t)\end{aligned}$$

证毕. □

注意所使用的 B 的性质仅仅是反对称性, 因而, 若 $\frac{\mathrm{d}}{\mathrm{d}t}T = BT - TB$ 且 $B^{\mathrm{T}} = -B$, 则对一切 t, $T(t)$ 有相同的特征值.

定理5.17 当 $t \to +\infty$ 或 $t \to -\infty$ 时, $T(t)$ 收敛于一个特征值在对角线上的对角阵.

证明 要证明当 $t \to \pm\infty$ 时, $b_i(t) \to 0$. 通过证明 $\int_{-\infty}^{\infty} \sum_{i=1}^{n-1} b_i^2(t)\mathrm{d}t < \infty$ 入手. 用归纳法证明 $\int_{-\infty}^{\infty}(b_j^2(t) + b_{n-j}^2(t))\mathrm{d}t < \infty$ 然后对一切 j 相加这些不等式. 当 $j = 0$ 时, 由假设可得 $\int_{-\infty}^{\infty}(b_0^2(t) + b_n^2(t))\mathrm{d}t$ 为 0.

现在设 $\varphi(t) = a_j(t) - a_{n-j+1}(t)$. 对一切 t, $\varphi(t)$ 是以 $2\|T(t)\|_2 = 2\|T(0)\|_2$ 为界. 于是

$$\begin{aligned}\dot{\varphi}(t) = \dot{a}_j(t) - \dot{a}_{n-j+1}(t) &= 2(b_j^2(t) - b_{j-1}^2(t)) - 2(b_{n-j+1}^2(t) - b_{n-j}^2(t))\\ &= 2(b_j^2(t) + b_{n-j}^2(t)) - 2(b_{j-1}^2(t) + b_{n-j+1}^2(t))\end{aligned}$$

故

$$\begin{aligned}\varphi(\tau) - \varphi(-\tau) &= \int_{-\tau}^{\tau}\dot{\varphi}(t)\mathrm{d}t\\ &= 2\int_{-\tau}^{\tau}(b_j^2(t) + b_{n-j}^2(t))\mathrm{d}t - 2\int_{-\tau}^{\tau}(b_{j-1}^2(t) + b_{n-j+1}^2(t))\mathrm{d}t.\end{aligned}$$

由归纳假设对一切 τ 最后的积分有界且对一切 τ, $\varphi(\tau) - \varphi(-\tau)$ 也有界, 故

$\int_{-\infty}^{\infty}(b_j^2(t)+b_{n-j}^2(t))\mathrm{d}t$ 必有界.

设 $p(t)=\sum_{i=1}^{n-1}b_i^2(t)$. 现在知道 $\int_{-\infty}^{\infty}p(t)\mathrm{d}t<\infty$，并且因为 $p(t)\geqslant 0$，所以要断

256

定 $\lim_{t\to\pm\infty}p(t)=0$. 但是当 $t\to\pm\infty$ 时需要把 $p(t)$ 有狭窄的波峰的可能性排除在外，如果设有 $p(t)$ 逼近于0，此时 $\int_{-\infty}^{\infty}p(t)\mathrm{d}t$ 可能有限. 通过证明 $p(t)$ 的导数有界来证明 $p(t)$ 没有波峰：

$$|\dot{p}(t)|=\left|\sum_{i=1}^{n-1}2\dot{b}_i(t)b_i(t)\right|=\left|\sum_{i=1}^{n-1}2b_i^2(t)(a_{i+1}(t)-a_i(t))\right|\leqslant 4(n-1)\|T\|_2^3.\ \square$$

因此，原则上可以利用关于 Toda 流率的一个 ODE 解算器求解特征值问题，但这个方法不快于其他现有的方法. 对 Toda 流率的兴趣在于它与 QR 算法的密切关系.

定义 5.5　设 $X_$ 表示 X 的严格下三角部分，而 $\pi_0(X)=X_-X_^{\mathrm{T}}$.

注意 $\pi_0(X)$ 是反对称的，并且若 X 已经是反对称的，则 $\pi_0(X)=X$. 因而 π_0 射影到反对称矩阵上.

考虑微分方程

$$\frac{\mathrm{d}}{\mathrm{d}t}T=BT-TB,\tag{5.28}$$

其中 $B=-\pi_0(F(T))$ 而 F 是从实数到实数的任意光滑函数. 因为 $B=-B^{\mathrm{T}}$，所以定理 5.16 证明对所有的 t，$T(t)$ 有相同的特征值. 选择 $F(x)=x$ 对应于我们刚才研究过的 Toda 流率，因为此时

$$-\pi_0(F(T))=-\pi_0(T)=\begin{bmatrix}0 & b_1 & & \\ -b_1 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1}\\ & & -b_{n-1} & 0\end{bmatrix}=B.$$

下面的定理建立 QR 分解与微分方程(5.28)的关系.

定理 5.18　设 $F(T(0))=F_0$. 设 $e^{tF_0}=Q(t)R(t)$ 是 QR 分解. 则 $T(t)=Q^{\mathrm{T}}(t)T(0)Q(t)$ 是方程(5.28)的解.[1]

把本定理的证明延迟到后面进行. 若我们正确地选择函数 F，它证实用 QR 迭代(算法 4.4)计算的迭代等同(identical)于微分方程的解.

定义 5.6　在方程(5.28)中选择 $F(x)=\log x$ 得到一个称为 QR 流率的微分方程.

推论 5.3　设 $F(x)=\log x$. 假定 $T(0)$ 正定，故 $\log T(0)$ 是实的. 设 $T_0\equiv T(0)=QR$，$T_1=RQ$ 等等是用无位移的 QR 迭代产生的矩阵序列. 则 $T(i)=T_i$. 因此 QR 算法给

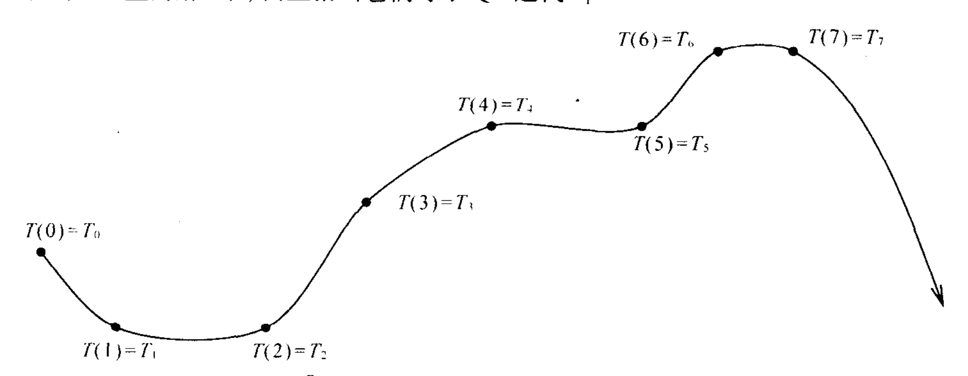
---

1. 注意 QR 分解不是完全唯一的(Q 可以用 QS 代替，R 可用 SR 代替，其中 S 是一个对角元为 ±1 的对角矩阵)所以 T_i 和 $T_{(i)}$ 实际上可能相差一个相似性 $T_i=ST(i)S^{-1}$，为简单起见，在这里以及在推论 5.4 中选择 S 使 $T_i=T(i)$.

出 QR 流率在整数时间 t 上的解.

推论的证明 在 $t=1$ 时，我们得到 $e^{t\log T_0}=T_0=Q(1)R(1)$，这是 T_0 的 QR 分解，而且正如所求的那样$T(1)=Q^T(1)T_0Q(1)=R(1)Q(1)=T_1$. 因为 ODE 的解唯一，所以对较大的 i 可提供证明 $T(i)=T_i$. □ [257]

下列图形生动地说明这个推论. 曲线表示微分方程的解. 点表示在整数时间 $t=0, 1, 2\cdots$上的解 $T(i)$而且指出它们等于 QR 迭代 T_i.



定理 5.18 的证明 微分 $e^{tF_0}=QR$ 得到

$$F_0e^{tF_0}=\dot{Q}R+Q\dot{R}$$

$$\text{或 } \dot{Q}=F_0e^{tF_0}R^{-1}-Q\dot{R}R^{-1}$$

$$\begin{aligned}\text{或 } Q^T\dot{Q}&=Q^TF_0e^{tF_0}R^{-1}-\dot{R}R^{-1}\\&=Q^TF_0(QR)R^{-1}-\dot{R}R^{-1} &&\text{因为 } e^{tF_0}=QR\\&=Q^TF(T(0))Q-\dot{R}R^{-1} &&\text{因为 } F_0=F(T(0))\\&=F(Q^TT(0)Q)-\dot{R}R^{-1}\\&=F(T)-\dot{R}R^{-1}.\end{aligned}$$

因为 $I=Q^TQ$ 推得 $0=\frac{d}{dt}Q^TQ=\dot{Q}^TQ+Q^T\dot{Q}=(Q^T\dot{Q})^T+(Q^T\dot{Q})$. 这意味着 $Q^T\dot{Q}$ 是反对称的，故$\pi_0(Q^T\dot{Q})=Q^T\dot{Q}=\pi_0(F(T)-\dot{R}R^{-1})$. 因为 $\dot{R}R^{-1}$是上三角阵，所以它不影响π_0，因而最后 $Q^T\dot{Q}=\pi_0(F(T))$.

$$\begin{aligned}\frac{d}{dt}T(t)&=\frac{d}{dt}Q^T(t)T(0)Q(t)\\&=\dot{Q}^TT(0)Q+Q^TT(0)\dot{Q}\\&=\dot{Q}^T(QQ^T)T(0)Q+Q^TT(0)(QQ^T)\dot{Q}\\&=\dot{Q}^TQT(t)+T(t)Q^T\dot{Q}\\&=-Q^T\dot{Q}T(t)+T(t)Q^T\dot{Q}\\&=-\pi_0(F(T(t)))T(t)+T(t)\pi_0(F(T(t)))\end{aligned}$$

证毕. □

下面的推论说明问题4.15中观察的现象，其中QR可“逆向运行”作出并返回

258

到初始矩阵. 也可见问题5.25.

推论5.4 假定通过下列步骤从正定阵 $\boldsymbol{T}_0$ 得到 $\boldsymbol{T}_4$：

1. 对 $\boldsymbol{T}_0$ 做 m 步无位移QR算法得到 $\boldsymbol{T}_1$.
2. 设 $\boldsymbol{T}_2$ = “翻转 $\boldsymbol{T}_1$” $=\boldsymbol{J}\boldsymbol{T}_1\boldsymbol{J}$，其中 $\boldsymbol{J}$ 等于把单位阵的列反序后的矩阵.
3. 对 $\boldsymbol{T}_2$ 做 m 步无位移的QR算法得到 $\boldsymbol{T}_3$.
4. 设 $\boldsymbol{T}_4=\boldsymbol{J}\boldsymbol{T}_3\boldsymbol{J}$.

则 $\boldsymbol{T}_4=\boldsymbol{T}_0$.

证明 若 $\boldsymbol{X}=\boldsymbol{X}^{\mathrm{T}}$，则容易验证 $\pi_0(\boldsymbol{J}\boldsymbol{X}\boldsymbol{J})=-\boldsymbol{J}\pi_0(\boldsymbol{X})\boldsymbol{J}$，故 $\boldsymbol{T}_J(t)\equiv\boldsymbol{J}\boldsymbol{T}(t)\boldsymbol{J}$ 满足

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{T}_J(t)&=\boldsymbol{J}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{T}(t)\boldsymbol{J}\\
&=\boldsymbol{J}[-\pi_0(F(\boldsymbol{T}))\boldsymbol{T}+\boldsymbol{T}\pi_0(F(\boldsymbol{T}))]\boldsymbol{J}\\
&=-\boldsymbol{J}\pi_0(F(\boldsymbol{T}))\boldsymbol{J}(\boldsymbol{J}\boldsymbol{T}\boldsymbol{J})+(\boldsymbol{J}\boldsymbol{T}\boldsymbol{J})\boldsymbol{J}\pi_0(F(\boldsymbol{T}))\boldsymbol{J},\text{因为 }\boldsymbol{J}^2=\boldsymbol{I}\\
&=\pi_0(\boldsymbol{J}F(\boldsymbol{T})\boldsymbol{J})\boldsymbol{T}_J-\boldsymbol{T}_J\pi_0(\boldsymbol{J}F(\boldsymbol{T})\boldsymbol{J})\\
&=\pi_0(F(\boldsymbol{J}\boldsymbol{T}\boldsymbol{J}))\boldsymbol{T}_J-\boldsymbol{T}_J\pi_0(F(\boldsymbol{J}\boldsymbol{T}\boldsymbol{J}))\\
&=\pi_0(F(\boldsymbol{T}_J))\boldsymbol{T}_J-\boldsymbol{T}_J\pi_0(F(\boldsymbol{T}_J)).
\end{aligned}
$$

这几乎是与 $\boldsymbol{T}(t)$ 一样相同的方程. 事实上，它精确地满足与 $\boldsymbol{T}(-t)$ 一样的方程：

$$
\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{T}(-t)=-\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{T}\bigg|_{-t}=-[\pi_0(F(\boldsymbol{T}))\boldsymbol{T}+\boldsymbol{T}\pi_0(F(\boldsymbol{T}))]_{-t}.
$$

因而用相同的初始条件 $\boldsymbol{T}_2$，$\boldsymbol{T}_J(t)$ 和 $\boldsymbol{T}(-t)$ 必定相等. 对时间 t 积分，$\boldsymbol{T}(-t)$ 取 $\boldsymbol{T}_2=\boldsymbol{J}\boldsymbol{T}_1\boldsymbol{J}$ 恢复到初始状态 $\boldsymbol{J}\boldsymbol{T}_0\boldsymbol{J}$，故 $\boldsymbol{T}_3=\boldsymbol{J}\boldsymbol{T}_0\boldsymbol{J}$ 而 $\boldsymbol{T}_4=\boldsymbol{J}\boldsymbol{T}_3\boldsymbol{J}=\boldsymbol{T}_0$. 证毕. □

5.5.2 与偏微分方程的关系

首次阅读本节可以跳过.

设 $\boldsymbol{T}(t)=-\dfrac{\partial^2}{\partial x^2}+q(x,t)$ 和 $\boldsymbol{B}(t)=-4\dfrac{\partial^3}{\partial x^3}+3\left(q(x,t)\dfrac{\partial}{\partial x}+\dfrac{\partial}{\partial x}q(x,t)\right)$. $\boldsymbol{T}(t)$ 和 $\boldsymbol{B}(t)$ 都是函数的线性算子，即矩阵的推广.

代入到 $\dfrac{\mathrm{d}\boldsymbol{T}}{\mathrm{d}t}=\boldsymbol{B}\boldsymbol{T}-\boldsymbol{T}\boldsymbol{B}$，倘若我们对 q 选择恰当的边界条件，得到

$$
q_t=6qq_x-q_{xxx}. \tag{5.29}
$$

259

($\boldsymbol{B}$ 必定是反对称的，而 $\boldsymbol{T}$ 是对称的.) 方程(5.29)称为Kortewegde Vries方程，它描述浅沟中的水流. 可以严格地证明在对一切 t，ODE

$$
\left(-\frac{\partial^2}{\partial x^2}+q(x,t)\right)h(x)=\lambda h(x)
$$

有某个特征值$\lambda_1,\lambda_2,\cdots$的无限集的意义下(5.29)对一切$t$保持$T(t)$的特征值. 换言之, 存在一个类能量 (energylike quantity) 的无限序列通过 Korteweg-de Vries 方程守恒. 这在理论上和数值上都是重要的.

关于 Toda 流率更多的细节见[144,170,67,68,239]和[188]中 Kruskal[166], Flaschka[106]和 Moser[187]的论文.

5.6 第5章参考书目和其他话题

对称特征问题的一本卓越的、全面的参考书是[197]. 关于相对扰动理论的材料可以在[75,82,101]中找到. 5.2.1节是以后面这些参考书目为基础的. 有关的工作在[66,92,228,250]中找到. 关于一般线性算子的扰动理论的经典教材是[161], 关于对称特征问题并行算法的综述见[76]. 求双对角阵 SVD 的 QR 算法在[80,67,120]中讨论. 而 dqds 算法是在[104,200,209]中讨论. 对分法的误差分析见[73,74,156], 加速对分法的最近的尝试见[105,203,201,176,173,175,269]. 改进逆迭代的最近的工作出现在[105,83,201,203]中. 分而治之特征程序是在[59]中引入的, 进一步讨论在[13,90,127,131,153,172,210,234]中. 从雅可比法得到高精度特征值的可能性在[66,75,82,92,183,228]中讨论. *Toda* 流率和相关的现象在[67,68,106,144,166,170,187,188,239]中讨论.

5.7 第5章问题

问题5.1(容易, Z. Bai) 证明$A=B+iC$是埃尔米特阵当且仅当

$$M=\begin{bmatrix} B & -C \\ C & B \end{bmatrix}$$

对称. 根据A的特征值和特征向量来表达M的特征值和特征向量.

问题5.2(中等) 利用外尔定理(定理5.1)和定理3.3的第4部分证明推论5.1.

问题5.3(中等) 考察图5.1. 对具有特征值$\alpha_3\leqslant\alpha_2\leqslant\alpha_1$的任意$3\times3$阶矩阵$A$考察相应的周线图. 设$C_1$和$C_2$是沿着$\rho(u,A)=\alpha_2$的两个大圆. 它们相交成什么角?

260

问题5.4(困难) 利用柯朗-费希尔极小极大定理(定理5.2)证明柯西交错定理(Cauchy interlace theorem);

- 假定$A=\begin{bmatrix} H & b \\ b^{\mathrm{T}} & u \end{bmatrix}$是$n\times n$阶对称矩阵而$H$是$(n-1)\times(n-1)$阶矩阵. 设$\alpha_n\leqslant\cdots\leqslant\alpha_1$是$A$的特征值而$\theta_{n-1}\leqslant\cdots\leqslant\theta_1$是$H$的特征值. 证明这两组特征值交错(interlace):

$$\alpha_n \leq \theta_{n-1} \leq \cdots \leq \theta_i \leq \alpha_i \leq \theta_{i-1} \leq \alpha_{i-1} \leq \cdots \leq \theta_1 \leq \alpha_1.$$

- 设 $A = \begin{bmatrix} H & B \\ B^T & U \end{bmatrix}$ 是 $n \times n$ 阶矩阵而 H 是 $m \times m$ 阶矩阵具有特征值 $\theta_m \leq \cdots \leq \theta_1$. 证明 A 和 H 的特征值在 $\alpha_{j+(n-m)} \leq \theta_j \leq \alpha_j$（或等价地 $\alpha_j \leq \theta_{j-(n-m)} \leq \alpha_{j-(n-m)}$）意义下交错.

问题5.5（中等）　设 $A = A^T$ 有特征值 $\alpha_1 \geq \cdots \geq \alpha_n$. $H = H^T$ 有特征值 $\theta_1 \geq \cdots \geq \theta_n$. 设 $A+H$ 有特征值 $\lambda_1 \geq \cdots \geq \lambda_n$. 利用柯朗－费希尔极小极大定理（定理5.2）证明 $\alpha_j + \theta_n \leq \lambda_j \leq \alpha_j + \theta_1$. 若 H 正定，则推断 $\lambda_j > \alpha_j$. 换言之，把一个对称正定阵 H 加到另一个对称阵 A 上只能增加它的特征值.

这个结果将用于定理7.1的证明.

问题5.6（中等）　设 $A = [A_1, A_2]$ 是 $n \times n$ 阶矩阵，其中 A_1 是 $n \times m$ 阶矩阵而 A_2 是 $n \times (n-m)$ 阶矩阵. $\sigma_1 \geq \cdots \geq \sigma_n$ 是 A 的奇异值而 $\tau_1 \geq \cdots \geq \tau_m$ 是 A_1 的奇异值. 利用问题5.4的柯西交错定理和定理3.3的第4部分证明 $\sigma_j \geq \tau_j \geq \sigma_{j+n-m}$.

问题5.7（中等）　设 q 是一个单位向量而 d 是正交于 q 的任何向量. 证明 $\|(q+d)q^T - I\|_2 = \|q+d\|_2$.（这个结果用于定理5.4的证明）.

问题5.8（困难）　模仿定理5.4明确地表达和证明关于奇异向量的一个定理.

问题5.9（困难）　由定理5.5证明界(5.6).

问题5.10（较困难）　由定理5.5证明界(5.7).

问题5.11（容易）　假如 $\theta = \theta_1 + \theta_2$，其中所有三个角都位于0和$\pi/2$之间. 证明 $\frac{1}{2}\sin 2\theta \leq \frac{1}{2}\sin 2\theta_1 + \frac{1}{2}\sin 2\theta_2$. 这个结果用于定理5.7的证明.

261　**问题5.12**（困难）　证明推论5.2. 提示：利用定理3.3的第4部分.

问题5.13（中等）　设 A 是对称阵. 考虑在每次迭代用瑞利商位移（$\sigma_i = a_{nn}$）运行位移的QR迭代（算法4.5），得到一个位移序列 $\sigma_1, \sigma_2, \cdots$. 从 $x_0 = [0, \cdots, 0, 1]^T$ 出发再运行瑞利商迭代（算法5.1），得到瑞利商序列 $\rho_1, \rho_2 \cdots$. 证明这些序列是恒等的：对一切 i，$\sigma_i = \rho_i$. 这证实5.3.2节中位移的QR迭代享有局部立方收敛的断言是成立的.

问题5.14（容易）　证明引理5.1.

问题5.15（容易）　证明：若 $t(n) = 2t(n/2) + cn^3 + O(n^2)$，则 $t(n) \approx c\frac{4}{3}n^3$. 这证明了分而治之算法（算法5.2）的复杂性分析是成立的.

问题5.16（容易）　设 $A = D + \rho uu^T$，其中 $D = \mathrm{diag}(d_1, \cdots, d_n)$ 而 $u = [u_1, \cdots, u_n]^T$. 证明：若 $d_i = d_{i+1}$ 或 $u_i = 0$，则 d_i 是 A 的一个特征值. 若 $u_i = 0$，证明对应于 d_i 的特征向量是单位阵的第 i 列 e_i. 当 $d_i = d_{i+1}$ 时导出一个类似的简单表达式. 这表明在分而治之算法（算法5.2）中如何处理收缩.

问题5.17(容易) 设ψ和ψ'是给定的标量. 说明如何计算函数定义$h(\lambda)=\hat{c}+\frac{c}{d-\lambda}$中的标量$c$和$\hat{c}$使得在$\lambda=\xi$上, $h(\xi)=\psi$和$h'(\xi)=\psi'$. 这个结果对导出5.3.3节中的特征方程解算器是需要的.

问题5.18(容易, Z. Bai) 利用SVD证明: 若A是$m\times n$阶实阵, $m\geqslant n$, 则存在一个$m\times n$阶矩阵Q, 具有规范正交列($Q^{\mathrm{T}}Q=I$)和一个$n\times n$阶半正定阵P使得$A=QP$. 这个分解称为A的极分解(polar decomposition), 因为它类似于复数的极形式$z=\mathrm{e}^{\mathrm{i}\arg(z)}\cdot|z|$. 证明: 若$A$非奇异, 则极分解唯一.

问题5.19(容易) 证明引理5.5.

问题5.20(容易) 证明引理5.7.

问题5.21(困难) 证明定理5.13, 而且把定理5.13中的指数$4n-2$减少为$2n-1$. 提示: 在引理5.7中用一个适当选取的常数乘D_1和除D_2.

问题5.22(中等) 假设$G^{\mathrm{T}}G$收敛于对角阵, 证明算法5.13计算G的SVD. 262

问题5.23(较困难) 设A是$n\times n$阶对称正定阵, 具有楚列斯基分解$A=LL^{\mathrm{T}}$, 且设$\hat{L}$是用浮点算术运算计算的楚列斯基因子. 本题中将界定$\hat{L}$的(平方的)奇异值当作A的特征值近似的相对误差. 证明A可写成$A=D\bar{A}D$, 其中$D=\mathrm{diag}(a_{11}^{1/2},\cdots,a_{nn}^{1/2})$且对一切$i,\bar{a}_{ii}=1$. 记$L=DX$. 证明$\kappa^2(X)=\kappa(\bar{A})$. 利用楚列斯基分解$A+\delta A=\hat{L}\hat{L}^{\mathrm{T}}$的向后误差$\delta A$的界(2.16), 证明我们可记$\hat{L}^{\mathrm{T}}\hat{L}=Y^{\mathrm{T}}L^{\mathrm{T}}LY$, 其中$\|Y^{\mathrm{T}}Y-I\|_2\leqslant O(\varepsilon)\kappa(\bar{A})$. 利用定理5.6断定$\hat{L}^{\mathrm{T}}\hat{L}$和$L^{\mathrm{T}}L$的特征值的相对差别至多为$O(\varepsilon)\kappa(\bar{A})$. 然后证明这个结果对$\hat{L}\hat{L}^{\mathrm{T}}$和$LL^{\mathrm{T}}$的特征值也成立. 这意味着$\hat{L}$奇异值的平方与$A$的特征值相对差别至多为$O(\varepsilon)\kappa(\bar{A})=O(\varepsilon)\kappa^2(L)$.

问题5.24(较困难) 这个问题证实SVD的单边雅可比法(算法5.13)的停止准则是恰当的. 设$A=G^{\mathrm{T}}G$, 其中G和A是$n\times n$阶矩阵. 假如对一切$j\neq k$, $|a_{jk}|\leqslant\varepsilon\sqrt{a_{jj}a_{kk}}$. 设$\sigma_n\leqslant\cdots\leqslant\sigma_1$是$G$的奇异值, 而$\alpha_n^2\leqslant\cdots\leqslant\alpha_1^2$是$A$中挑出的对角元. 证明$|\sigma_i-\alpha_i|\leqslant n\varepsilon|\alpha_i|$以致$\alpha_i$等于奇异值达到高的相对精度. 提示: 利用推论5.2.

问题5.25(较困难) 在问题4.15中, "注意到"对对称矩阵运行m步QR迭代, "翻转"行和列, 运行另外的m步, 并再次翻转得到原来的矩阵. (翻转X意味着用JXJ代替X, 其中J是把单位矩阵的行反序后得到的矩阵). 在本练习中将利用不同于推论5.4的一个方法对对称阵证明这个结论.

考虑零位移的LR迭代(算法5.9), 应用于对称正定阵T(它不一定是三对角阵): 设$T=T_0=B_0^{\mathrm{T}}B_0$是楚列斯基分解, $T_1=B_0B_0^{\mathrm{T}}=B_1^{\mathrm{T}}B_1$, 以及更一般地, $T_i=B_{i-1}B_{i-1}^{\mathrm{T}}=B_i^{\mathrm{T}}B_i$. 设$\hat{T}$表示从$T_0$出发经无位移的QR迭代$i$步之后得到的矩阵; 即$\hat{T}_i=Q_iR_i$是QR分解, 然后$\hat{T}_{i+1}=R_iQ_i$. 在引理5.6中我们证明了$\hat{T}_i=T_{2i}$; 即QR一步和$LR$的两步相同.

1. 证明$T_i=(B_{i-1}B_{i-2}\cdots B_0)^{-\mathrm{T}}T_0(B_{i-1}B_{i-2}\cdots B_0)^{\mathrm{T}}$.

2. 证明 $\boldsymbol{T}_i=(\boldsymbol{B}_{i-1}\boldsymbol{B}_{i-2}\cdots\boldsymbol{B}_0)\boldsymbol{T}_0(\boldsymbol{B}_{i-1}\boldsymbol{B}_{i-2}\cdots\boldsymbol{B}_0)^{-1}$.

3. 证明 $\boldsymbol{T}_0^i=(\boldsymbol{B}_i\boldsymbol{B}_{i-1}\cdots\boldsymbol{B}_0)^{\mathrm{T}}(\boldsymbol{B}_i\boldsymbol{B}_{i-1}\cdots\boldsymbol{B}_0)$是 T_0^i 的楚列斯基分解.

4. 证明 $\boldsymbol{T}_0^i=(\boldsymbol{Q}_0\cdots\boldsymbol{Q}_{i-2}\boldsymbol{Q}_{i-1})\cdot(\boldsymbol{R}_{i-1}\boldsymbol{R}_{i-2}\cdots\boldsymbol{R}_0)$是 $\boldsymbol{T}_0^i$ 的 QR 分解.

[263] 5. 证明 $\boldsymbol{T}_0^{2i}=(\boldsymbol{R}_{2i-1}\boldsymbol{R}_{2i-2}\cdots\boldsymbol{R}_0)^{\mathrm{T}}(\boldsymbol{R}_{2i-1}\boldsymbol{R}_{2i-2}\cdots\boldsymbol{R}_0)$是 $\boldsymbol{T}_0^{2i}$ 的楚列斯基分解.

6. 证明QR 分解 m 步，翻转，QR 分解 m 步和翻转之后的结果和原矩阵相同. 提示：利用楚列斯基分解是唯一的事实.

问题 5.26（困难；Z. Bai）　假如 $\boldsymbol{x}$ 是一个 n 维向量. 由 $c_{ij}=|x_i|+|x_j|-|x_i-x_j|$定义矩阵 $\boldsymbol{C}$. 证明 $\boldsymbol{C}(x)$半正定.

问题 5.27（困难；Z. Bai）　设

$$\boldsymbol{A}=\begin{pmatrix}\boldsymbol{I} & \boldsymbol{B}\\ \boldsymbol{B}^H & \boldsymbol{I}\end{pmatrix}$$

和$\|\boldsymbol{B}\|_2<1$. 证明

$$\|\boldsymbol{A}\|_2\|\boldsymbol{A}^{-1}\|_2=\frac{1+\|\boldsymbol{B}\|_2}{1-\|\boldsymbol{B}\|_2}.$$

问题 5.28（中等；Z. Bai）　若 $\boldsymbol{A}^*=-\boldsymbol{A}$，则方阵 $\boldsymbol{A}$ 称为反埃尔米特阵（skew Hermitian）. 证明

1. 反埃尔米特阵的特征值为纯虚数.

2. $\boldsymbol{I}-\boldsymbol{A}$ 非奇异.

[264] 3. $\boldsymbol{C}=(\boldsymbol{I}-\boldsymbol{A})^{-1}(\boldsymbol{I}+\boldsymbol{A})$是酉阵. $\boldsymbol{C}$ 称为 $\boldsymbol{A}$ 的凯莱变换（Cayley transform）.

第6章　线性方程组迭代方法

6.1 概　　述

当诸如高斯消元法等方法需要太多的时间和空间时，我们就使用迭代算法求解 $\boldsymbol{Ax}=\boldsymbol{b}$. 诸如高斯消去法这些在有限步之后计算精确解(无舍入时!)的方法称为直接法. 与直接法相反，迭代法在有限步之后一般不产生精确解，但是在每步之后逐渐减少误差. 当误差小于用户提供的阈值时迭代停止. 最后的误差依赖于我们作了多少次迭代以及方法和线性方程组的性质. 我们的总体目标是开发一些方法，每次迭代都能较大幅度地减少误差并且计算量较少.

本领域中的许多工作涉及从基本的数学问题或物理问题引出线性方程组，进而设计出较好的迭代法. 基本的问题经常是涉及一个微分方程的物理系统的有限差分或有限元模型. 存在许多种类的物理系统、微分方程以及有限差分和有限元模型，因此有许多方法. 我们不奢求论述所有的或者甚至大多数感兴趣的情况，只限于研究正方形上泊松方程的标准有限差分近似的模型问题. 泊松方程及与其密切相关的拉普拉斯方程在许多应用中出现，例如包括电磁学、流体力学、热流、扩散和量子力学. 除了描述每个方法如何对泊松方程操作外，还将指出它是如何广泛适用的，并描述通常的变形.

本章的其余部分组织如下. 6.2 节描述本章所讨论的迭代法的在线帮助和软件. 6.3 节详细描述模型问题的公式表示. 6.4 节总结和比较本章中求解模型问题(几乎)所有迭代法的性能.

后面5节的方法大致按其对模型问题有效性程度递增次序加以描述. 6.5 节描述最基本的迭代法：雅可比、高斯-塞德尔、逐次超松弛及它们的变形. 6.6 节描述克雷洛夫子空间法，集中讨论共轭梯度法. 6.7 节描述快速傅里叶变换以及如何利用它求解模型问题. 6.8 节描述块循环约化. 最后，6.9 节讨论多重网格法，这是模型问题最快的算法. 多重网格法每个未知量只需 $O(1)$ 次运算，这是最优的工作量.

6.10 节描述区域分解法，这是把前面几节中介绍的较简单的方法组合起来求解比模型问题更为复杂问题的一族方法.

6.2 迭代法的在线(On-line)帮助

关于泊松方程，将有一个数值方法的简表，列举的方法明显地超过我们讨论的

其他所有方法. 但对其他的线性方程组而言，它并非是最优的方法(这就是为什么我们对它谈论如此之多!). 为帮助用户在这些可利用的许多方法中选择求解他们的方程组的最优的方法，可利用 NETLIB/templates 上的在线帮助. 其中包含本章讨论的大多数迭代法的一本小册子[24]和软件. 小册子有 PostScript(NETLIB/templates/templates. ps)和 HTML(NETLIB/templates/Templates. html)两种格式. 软件则有 Matlab,Fortran 和C++ 代码.

因为算法的实现把矩阵表示的细节与算法本身分开，所以使用单词模板(template)描述这本册子和软件. 特别地，*克雷洛夫子空间方法*(Krylov subspace method)(见6.6节)只要求用任意的一个向量 $\boldsymbol{z}$ 乘矩阵 $\boldsymbol{A}$ 的能力. 做这件事的最佳方法只依赖于 $\boldsymbol{A}$ 如何被表示，而不影响算法的组织. 换言之，矩阵-向量乘法在模版中称为"黑盒子". 提供这个黑盒子的实现是用户的职责.

关于特征值问题类似的模板项目也正在进行中. 关于迭代法其他新近的教科书是[15,136,214].

由于最具挑战性的实际问题来自微分方程比模型问题更难，线性方程组$\boldsymbol{Ax}=\boldsymbol{b}$必须"预处理"，或用以某种方式更易求解的等价方程组 $\boldsymbol{M}^{-1}\boldsymbol{Ax}=\boldsymbol{M}^{-1}\boldsymbol{b}$ 代替. 这在6.6.5节和6.10节中详细地加以讨论. 这些方法的许多实现，包括并行计算实现，可以在 http://www.mcs.anl.gov/petsc/petsc.html[232]上的软件包 PETSc 或 Portable Extensible Toolkit. for Scientific computing(科学计算的可移动、可扩张的工具箱)中得到.

266

6.3 泊松方程

6.3.1 一维泊松方程

从泊松方程的一维形式开始,

$$-\frac{d^2v(x)}{dx^2}=f(x),0<x<1, \tag{6.1}$$

其中$f(x)$是给定的函数而$v(x)$是要计算的未知函数. $v(x)$也必须满足边界条件[1] $v(0)=v(1)=0$. 通过尝试计算0和1之间 $N+2$ 个等距点 x_i 上的近似解来离散这个问题: $x_i=ih$，其中 $h=\frac{1}{N+1}$且$0\leqslant i\leqslant N+1$. 我们简记 $v_i=v(x_i)$和$f_i=f(x_i)$. 为把微分方程(6.1)转换成未知量 $v_1,\cdots,v_N$ 的线性方程，我们利用*有限差分*(finite difference)去逼近

$$\left.\frac{dv(x)}{dx}\right|_{x=(i-0.5)h}\approx\frac{v_i-v_{i-1}}{h},$$

1. 这些称为 Dirichlet 边界条件. 其他种类的边界条件也是可能的.

$$\left.\frac{dv(x)}{dx}\right|_{x=(i+0.5)h} \approx \frac{v_{i+1}-v_i}{h}.$$

这些近似式相减并除以 h 得到中心差分近似(centered difference approximation)

$$-\left.\frac{d^2v(x)}{dx^2}\right|_{x=x_i} = \frac{2v_i - v_{i-1} - v_{i+1}}{h^2} - \tau_i, \tag{6.2}$$

其中所谓截断误差(truncation error) τ_i 可以证明是 $O\left(h^2 \cdot \left\|\frac{d^4v}{dx^4}\right\|_\infty\right)$. 现在可改写在 $x=x_i$ 上的方程(6.1)为

$$-v_{i-1}+2v_i-v_{i+1}=h^2f_i+h^2\tau_i,$$

其中 $0<i<N+1$. 因为边界条件可推出 $v_0=v_{N+1}=0$, 所以我们有 N 个未知量 $v_1,\cdots v_N$ 的 N 个方程:

$$T_N \cdot \begin{bmatrix} v_1 \\ \vdots \\ \vdots \\ v_N \end{bmatrix} \equiv \begin{bmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ \vdots \\ \vdots \\ v_N \end{bmatrix} = h^2 \begin{bmatrix} f_1 \\ \vdots \\ \vdots \\ f_N \end{bmatrix} + h^2 \begin{bmatrix} \tau_1 \\ \vdots \\ \vdots \\ \tau_N \end{bmatrix} \tag{6.3}$$

或

$$T_N v = h^2 f + h^2 \bar{\tau}. \tag{6.4}$$

为求解这个方程, 我们将略去 $\bar{\tau}$, 因为它与 f 相比是小的, 得到

$$T_N \hat{v} = h^2 f. \tag{6.5}$$

(我们在后面界定误差 $v-\hat{v}$).

267

系数矩阵 T_N 在下面全部工作中扮演一个重要的角色, 所以将相当详细地考察它. 首先, 将计算它的特征值和特征向量. 可以容易地利用三角恒等式证实下列引理(见问题6.1).

引理 6.1 T_N 的特征值是 $\lambda_j = 2(1-\cos\frac{\pi j}{N+1})$. 特征向量是 z_j, 其中 $z_j(k)=\sqrt{\frac{2}{N+1}}\sin(jk\pi/(N+1))$. z_j 有单位 2-范数. 设 $Z=[z_1,\cdots,z_N]$ 是列为特征向量的正交阵, 且 $\Lambda = \mathrm{diag}(\lambda_1,\cdots,\lambda_N)$, 故可记 $T_N = Z\Lambda Z^T$.

图 6-1 是 $N=21$ 的 T_N 的特征值图表.

最大的特征值是 $\lambda_N = 2(1-\cos\pi\frac{N}{N+1})\approx 4$. 最小的特征值[1]是 λ_1, 对小的 i

$$\lambda_i = 2\left(1-\cos\frac{i\pi}{N+1}\right) \approx 2\left(1-\left(1-\frac{i^2\pi^2}{2(N+1)^2}\right)\right) = \left(\frac{i\pi}{N+1}\right)^2.$$

因此 T_N 是正定的, 对大的 N 有条件数 $\lambda_N/\lambda_1 \approx 4(N+1)^2/\pi^2$. 特征向量是在 $j=1$ 有最低频率和在 $j=N$ 有最高频率的正弦曲线, 对 $N=21$ 如图 6-2 所示.

1. 注意 λ_N 是最大的特征值而 λ_1 是最小的特征值, 与第 5 章的惯例相反.

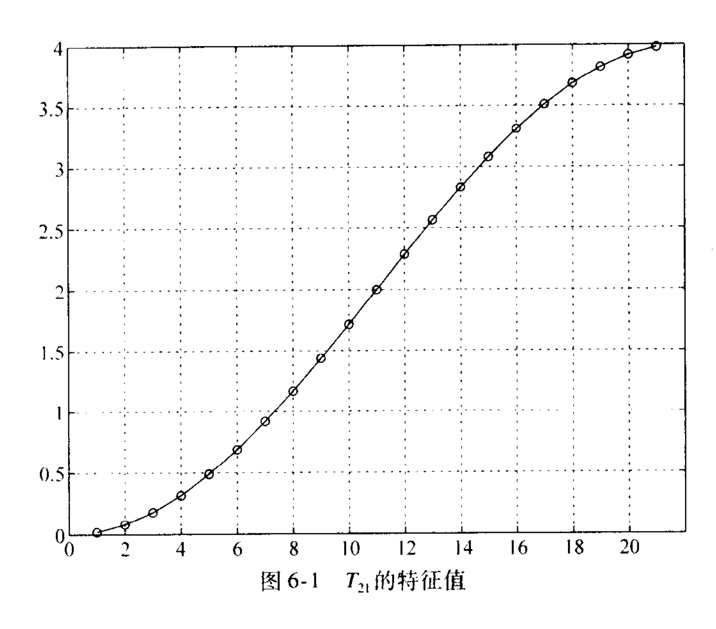


图 6-1 T_{21}的特征值

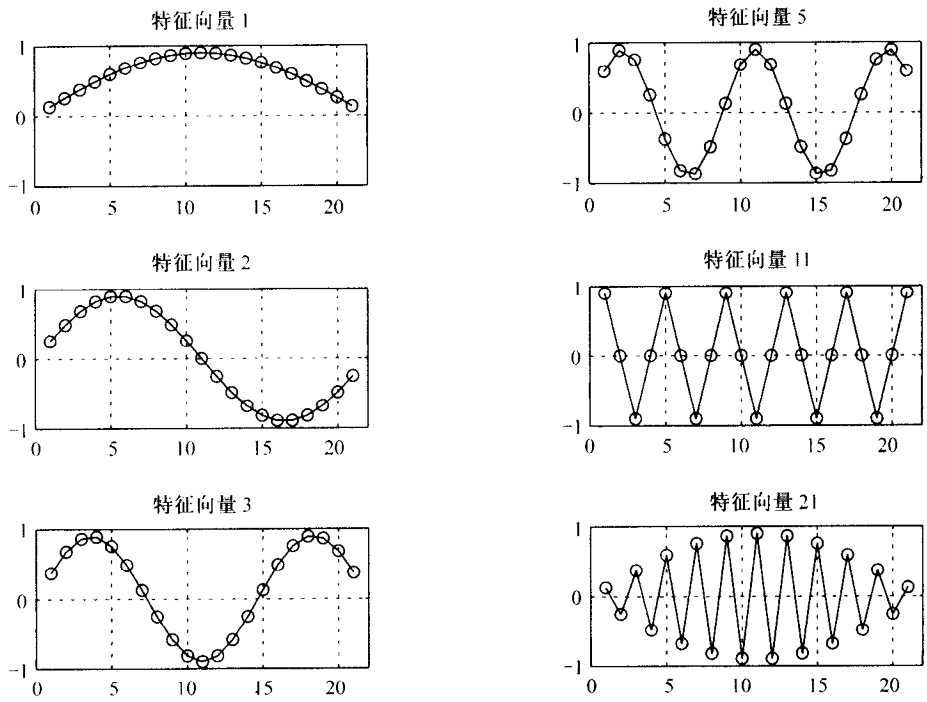


图 6-2 T_{21}的特征向量

现在足以看出误差界即 $T_N\hat{v}=h^2f$ 的解和微分方程的真解 v 之间差别：由方程(6.4)减方程(6.5)得到 $v-\hat{v}=h^2T_N^{-1}\tau$. 取范数得到

$$\|v-\hat{v}\|_2\leqslant h^2\|T_N^{-1}\|_2\|\tau\|_2\approx h^2\frac{(N+1)^2}{\pi^2}\|\tau\|_2=O(\|\tau\|_2)=O\left(h^2\left\|\frac{d^4v}{dx^4}\right\|_\infty\right),$$

倘若解足够光滑的话$\left(\left\|\frac{d^4v}{dx^4}\right\|_\infty\text{有界}\right)$，那么误差 $v-\hat{v}$ 与 h^2 成比例的趋向于零.

从现在开始我们将不区分 v 及其近似 $\hat{v}$，因而将用 $T_Nv=h^2f$ 简化记号.

除线性方程组 $h^{-2}T_Nv=f$ 的解逼近微分方程(6.1)的解之外，可以证实$h^{-2}T_N$的特征值和特征向量也逼近微分方程的特征值和特征函数(eigenfunction)：若

268
~
269

$$-\frac{d^2\hat{z}_i(x)}{dx^2}=\hat{\lambda}_i\hat{z}_i(x)\text{和}\hat{z}_i(0)=\hat{z}_i(1)=0.$$

则 $\hat{\lambda}_i$ 是微分方程的特征值而 $\hat{z}_i(x)$ 是微分方程的特征函数. 让我们求解 $\hat{\lambda}_i$ 和 $\hat{z}_i(x)$：容易看出对某个常数 α 和 β，$\hat{z}_i(x)$ 必定等于 $\alpha\sin\left(\sqrt{\hat{\lambda}_i}x\right)+\beta\cos\left(\sqrt{\hat{\lambda}_i}x\right)$. 边界条件 $\hat{z}_i(0)=0$ 推出 $\beta=0$，而边界条件 $\hat{z}_i(1)=0$ 推出$\sqrt{\hat{\lambda}_i}$ 是π的整数倍数，由此可取为 $i\pi$. 于是 $\hat{\lambda}_i=i^2\pi^2$，$\hat{z}_i(x)=\alpha\sin(i\pi x)$，对某个非零常数$\alpha$(可取$\alpha$为1). 因而特征向量 z_i 正好等于特征函数 $\hat{z}_i(x)$ 在样本点 $x_j=jh\left(\text{按}\sqrt{\frac{2}{N+1}}\text{比例调节时}\right)$上求值. 当 i 小的时候，用 $h^{-2}\cdot\lambda_i=(N+1)^2\cdot 2\left(1-\cos\frac{i\pi}{N+1}\right)=i^2\pi^2+O((N+1)^{-2})$ 充分地逼近 $\hat{\lambda}_i=i^2\pi^2$.

因此我们看出在 T_N(或 $h^{-2}T_N$)和二阶导数算子 $-\frac{d^2}{dx^2}$之间存在一个密切的对应. 这种对应将成为后面算法的设计和分析的动力.

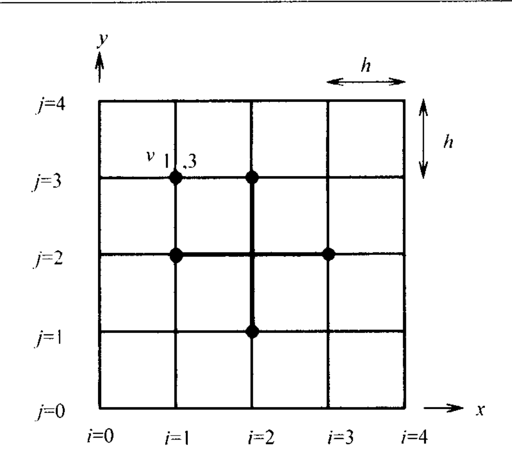
写下 T_N 的楚列斯基和LU因子的简单公式也是可能的. 细节见问题6.2.

6.3.2 二维泊松方程

现在转向二维泊松方程

$$-\frac{\partial^2 v(x,y)}{\partial x^2}-\frac{\partial^2 v(x,y)}{\partial y^2}=f(x,y) \tag{6.6}$$

在单位正方形$\{(x,y):0<x,y<1\}$上，连同在正方形边界上的边界条件 $v=0$. 在正方形内网格上，就是在(x_i,y_j)上离散化，$x_i=ih$，$y_j=jh$，$h=\frac{1}{N+1}$. 简写 $v_{ij}=v(ih,jh)$ 和 $f_{ij}=f(ih,jh)$，对 $N=3$ 如下面所示：



由方程(6.2)，我们可作近似

$$-\left.\frac{\partial^2 v(x,y)}{\partial x^2}\right|_{x=x_i,y=y_j} \approx \frac{2v_{i,j}-v_{i-1,j}-v_{i+1,j}}{h^2} \tag{6.7}$$

和

270

$$-\left.\frac{\partial^2 v(x,y)}{\partial y^2}\right|_{x=x_i,y=y_j} \approx \frac{2v_{i,j}-v_{i,j-1}-v_{i,j+1}}{h^2}. \tag{6.8}$$

把这些近似相加得到

$$\begin{aligned} & -\frac{\partial^2 v(x,y)}{\partial x^2}-\left.\frac{\partial^2 v(x,y)}{\partial y^2}\right|_{x=x_i,y=y_j} \\ & =\frac{4v_{ij}-v_{i-1,j}-v_{i+1,j}-v_{i,j-1}-v_{i,j+1}}{h^2}-\tau_{ij}, \end{aligned} \tag{6.9}$$

其中 τ_{ij} 还是以 $O(h^2)$ 为界的截断误差. 在上面的图形中间的粗的十字称为这个方程的(5点)型板(stencil)，因为它连接方程(6.9)中出现的所有5个 v 的值. 由边界条件知道 $v_{0j}=v_{N+1,j}=v_{i,0}=v_{i,N+1}=0$，因而(6.9)式确定一组具有 n 个未知量 $v_{ij},1\leqslant i,j\leqslant N$ 的 $n=N^2$ 个线性方程:

$$4v_{ij}-v_{i-1,j}-v_{i+1,j}-v_{i,j-1}-v_{i,j+1}=h^2 f_{ij}. \tag{6.10}$$

有两种方式改写(6.10)表示的 n 个方程为一个单独的矩阵方程，这两种方式将在后面使用.

第一种方式是把未知量 v_{ij} 作为元素 v_{ij} 填满 $N\times N$ 阶矩阵 $\boldsymbol{V}$ 而右端 $h^2 f_{ij}$ 类似地填满 $N\times N$ 阶矩阵 h^2F. 诀窍是依据 $\boldsymbol{V}$ 和 $\boldsymbol{T}_N$ 以一个简单的方式写出 i,j 元素为 $4v_{ij}-v_{i-1,j}-v_{i+1,j}-v_{i,j-1}-v_{i,j+1}$ 的矩阵：只要注意

$$2v_{ij}-v_{i-1,j}-v_{i+1,j}=(\boldsymbol{T}_N\cdot\boldsymbol{V})_{ij},$$
$$2v_{ij}-v_{i,j-1}-v_{i,j+1}=(\boldsymbol{V}\cdot\boldsymbol{T}_N)_{ij},$$

因而这两个式子相加得到

$$(\boldsymbol{T}_N\cdot\boldsymbol{V}+\boldsymbol{V}\cdot\boldsymbol{T}_N)_{ij}=4v_{ij}-v_{i-1,j}-v_{i+1,j}-v_{i,j-1}-v_{i,j+1}=h^2 f_{ij}=(h^2\boldsymbol{F})_{ij}$$

或

$$T_N \cdot V + V \cdot T_N = h^2 F. \tag{6.11}$$

这是一个矩阵 V 的未知元素的一个线性方程组，虽然它不是写成通常的“$Ax = b$”形式，和未知量构成一个向量 x.（在下面将写成“$Ax = b$”形式.）它仍然足以告诉我们在潜在的矩阵 A 的特征值和特征向量是什么，因为“$Ax = \lambda x$”和“$T_N V + V T_N = \lambda V$”是相同的. 现在假定 $T_N z_i = \lambda_i z_i$ 和 $T_N z_j = \lambda_j z_j$ 是 T_N 的任意两个特征对，且设 $V = z_i z_j^{\mathrm{T}}$. 则

$$\begin{aligned} T_N V + V T_N &= (T_N z_i) z_j^{\mathrm{T}} + z_i (z_j^{\mathrm{T}} T_N) \\ &= (\lambda_i z_i) z_j^{\mathrm{T}} + z_i (z_j^{\mathrm{T}} \lambda_j) \\ &= (\lambda_i + \lambda_j) z_i z_j^{\mathrm{T}} \\ &= (\lambda_i + \lambda_j) V, \end{aligned} \tag{6.12}$$

271

因此 $V = z_i z_j^{\mathrm{T}}$ 是一个“特征向量”而 $\lambda_i + \lambda_j$ 是一个特征值. 因为 V 有 N^2 个元素，所以对于 T_N 的每一对特征值 λ_i 和 λ_j 预期有 N^2 个特征值和特征向量. 特别地，最小的特征值是 $2\lambda_1$ 和最大的特征值是 $2\lambda_N$，因而条件数与一维情况相同. 下面利用“$Ax = b$”形式再次导出这个结果. 用 $z_i z_j^{\mathrm{T}}$ 的矩阵元素所定义的曲面表示的某些特征向量的图见图 6-3.

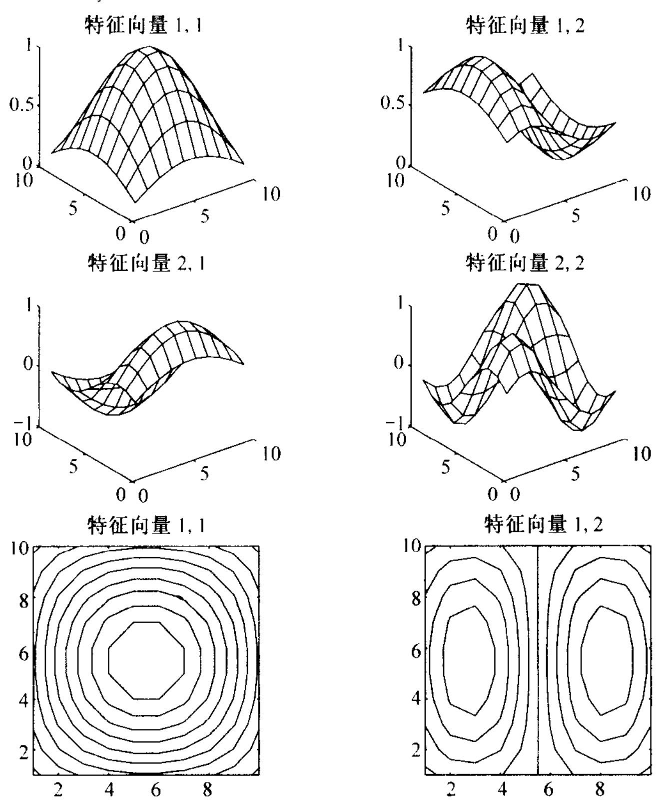
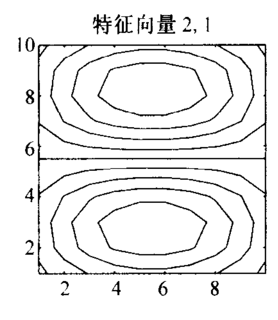


图 6-3 10 × 10 泊松方程的前四个特征向量的三维图和周线图



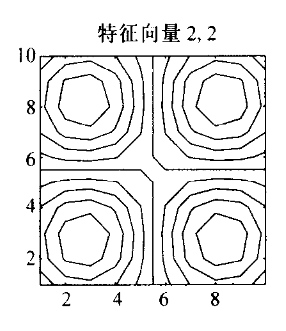


图6-3 (续)

正如 $h^{-2}\boldsymbol{T}_N$ 的特征值和特征向量是一维泊松方程的特征值和特征函数的优良的近似一样，对二维泊松方程同样成立，其特征值和特征函数为下面形式(见问题6.3)：

$$\left(-\frac{\partial^2}{\partial x^2}-\frac{\partial^2}{\partial y^2}\right)\sin(i\pi x)\sin(j\pi y)=(i^2\pi^2+j^2\pi^2)\sin(i\pi x)\sin(j\pi y). \tag{6.13}$$

把(6.10)式表示的 n 个方程写成一个单独的矩阵方程的第二种方式是把未知量 $\boldsymbol{v}_{ij}$ 写成一个单独的长度为 $N^2\times 1$ 的向量. 这需要对 v_{ij} 选择一个次序. 如图6-4所示我们(有些任意地)选择从左上方到右下方逐列对它们编号.

例如，当 $N=3$ 时得到一个列向量 $\boldsymbol{v}\equiv[v_1,\cdots,v_9]^{\mathrm{T}}$. 若我们按照它对 f 编号，则可变换(6.10)式得到

$$\boldsymbol{T}_{3\times 3}\cdot\begin{bmatrix}v_1\\ v_2\\ \vdots\\ \vdots\\ \vdots\\ v_9\end{bmatrix}\equiv\left[\begin{array}{ccc|ccc|ccc}4&-1&&-1&&&&&\\-1&4&-1&&-1&&&&\\&-1&4&&&-1&&&\\\hline -1&&&4&-1&&-1&&\\&-1&&-1&4&-1&&-1&\\&&-1&&-1&4&&&-1\\\hline &&&-1&&&4&-1&\\&&&&-1&&-1&4&-1\\&&&&&-1&&-1&4\end{array}\right]\begin{bmatrix}v_1\\ v_2\\ \vdots\\ \vdots\\ \vdots\\ v_9\end{bmatrix}=h^2\begin{bmatrix}f_1\\ f_2\\ \vdots\\ \vdots\\ \vdots\\ f_9\end{bmatrix} \tag{6.14}$$

紧邻对角线的 -1 对应于减去上部和下部邻近的值 $-v_{i,j-1}-v_{i,j+1}$. 远离对角线的 -1 对应于减去左面和右面邻近的值 $-v_{i-1,j}-v_{i+1,j}$. 对一般的 N，在下节中将证实能得到一个 $N^2\times N^2$ 线性方程组

272 ~ 273

$$\boldsymbol{T}_{N\times N}\cdot\boldsymbol{v}=h^2 f, \tag{6.15}$$

其中 $\boldsymbol{T}_{N\times N}$ 在对角线上有 N 个形状为 $\boldsymbol{T}_N+2\boldsymbol{I}_N$ 的 $N\times N$ 块矩阵而在非对角线上具有形状为 $-\boldsymbol{I}_N$ 的块矩阵：

$$\boldsymbol{T}_{N\times N}=\begin{bmatrix}\boldsymbol{T}_N+2\boldsymbol{I}_N & -\boldsymbol{I}_N & & \\ -\boldsymbol{I}_N & \ddots & \ddots & \\ & \ddots & \ddots & -\boldsymbol{I}_N\\ & & -\boldsymbol{I}_N & \boldsymbol{T}_N+2\boldsymbol{I}_N\end{bmatrix}. \tag{6.16}$$

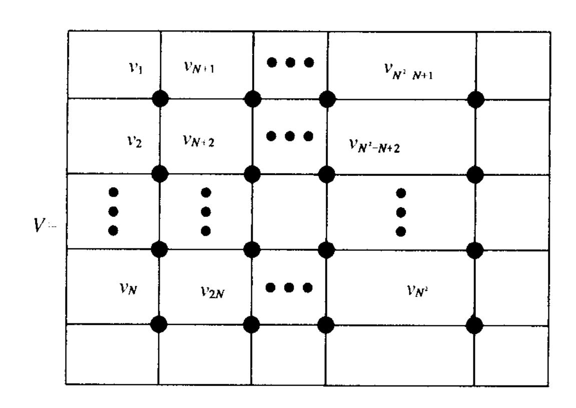


图 6-4 泊松方程中未知量编号

6.3.3 用克罗内克积表达泊松方程

下面是推导等式(6.15)和(6.16)同时又计算 $T_{N\times N}$的特征值和特征向量的一个系统方法. 对三维或更高维的泊松方程这个方法同样相当有效.

定义 6.1 设 $\boldsymbol{X}$ 是 $m\times n$ 阶矩阵. 则 $\mathrm{vec}(\boldsymbol{X})$ 是把 $\boldsymbol{X}$ 的列从左到右一列堆积在后面一列顶上构筑成维数大小为 $m\cdot n$ 的一个列向量.

注意图 6-4 中定义的 $N^2\times 1$ 向量 $\boldsymbol{v}$ 也可写成 $\boldsymbol{v}=\mathrm{vec}(\boldsymbol{V})$.

为表达 $\boldsymbol{T}_{N\times N}$同时又计算其特征值和特征向量, 需要列入克罗内克积(Kronecker product).

定义 6.2 设 $\boldsymbol{A}$ 是 $m\times n$ 阶矩阵而 $\boldsymbol{B}$ 是 $p\times q$ 阶矩阵, 则 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的克罗内克积 $\boldsymbol{A}\otimes\boldsymbol{B}$ 是$(m\cdot p)\times(n\cdot q)$阶矩阵

$$\begin{bmatrix} a_{1,1}\cdot B & \cdots & a_{1,n}\cdot B \\ \vdots & & \vdots \\ a_{m,1}\cdot B & \cdots & a_{m,n}\cdot B \end{bmatrix}.$$

下列引理告诉我们如何根据克罗内克积和 vec(·)算子改写泊松方程. [274]

引理 6.2 设 $\boldsymbol{A}$ 是 $m\times m$ 阶矩阵, $\boldsymbol{B}$ 是 $n\times n$ 阶矩阵而 $\boldsymbol{X}$ 和 $\boldsymbol{C}$ 是 $m\times n$ 阶矩阵. 则下列性质成立:

1. $\mathrm{vec}(\boldsymbol{AX})=(\boldsymbol{I}_n\otimes\boldsymbol{A})\cdot\mathrm{vec}(\boldsymbol{X})$.
2. $\mathrm{vec}(\boldsymbol{XB})=(\boldsymbol{B}^{\mathrm{T}}\otimes\boldsymbol{I}_m)\cdot\mathrm{vec}(\boldsymbol{X})$.
3. 泊松方程 $\boldsymbol{T}_N\boldsymbol{V}+\boldsymbol{V}\boldsymbol{T}_N=h^2\boldsymbol{F}$ 等价于

$$\boldsymbol{T}_{N\times N}\cdot\mathrm{vec}(\boldsymbol{V})\equiv(\boldsymbol{I}_N\otimes\boldsymbol{T}_N+\boldsymbol{T}_N\otimes\boldsymbol{I}_N)\cdot\mathrm{vec}(\boldsymbol{V})=h^2\mathrm{vec}(\boldsymbol{F}). \quad (6.17)$$

证明 只证明第 3 部分, 其余部分留作问题 6.4. 从(6.11)式表达的泊松方程$\boldsymbol{T}_N\boldsymbol{V}+\boldsymbol{V}\boldsymbol{T}_N=h^2\boldsymbol{F}$ 开始, 这个方程明显地等价于

$$\mathrm{vec}(\boldsymbol{T}_N\boldsymbol{V}+\boldsymbol{V}\boldsymbol{T}_N)=\mathrm{vec}(\boldsymbol{T}_N\boldsymbol{V})+\mathrm{vec}(\boldsymbol{V}\boldsymbol{T}_N)=\mathrm{vec}(h^2\boldsymbol{F}).$$

由引理的第 1 部分知

$$\mathrm{vec}(\boldsymbol{T}_N\boldsymbol{V}) = (\boldsymbol{I}_N \otimes \boldsymbol{T}_N)\,\mathrm{vec}(\boldsymbol{V}).$$

由引理的第 2 部分及 $\boldsymbol{T}_N$ 的对称性知

$$\mathrm{vec}(\boldsymbol{V}\boldsymbol{T}_N) = (\boldsymbol{T}_N^{\mathrm{T}} \otimes \boldsymbol{I}_N)\,\mathrm{vec}(\boldsymbol{V}) = (\boldsymbol{T}_N \otimes \boldsymbol{I}_N)\,\mathrm{vec}(\boldsymbol{V}).$$

把最后的两个表达式相加完成第 3 部分的证明. □

读者可以证实表达式

$$\boldsymbol{T}_{N\times N} = \boldsymbol{I}_N \otimes \boldsymbol{T}_N + \boldsymbol{T}_N \otimes \boldsymbol{I}_N = \begin{bmatrix} \boldsymbol{T}_N & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \boldsymbol{T}_N \end{bmatrix} + \begin{bmatrix} 2\boldsymbol{I}_N & -\boldsymbol{I}_N & & \\ -\boldsymbol{I}_N & \ddots & \ddots & \\ & \ddots & \ddots & -\boldsymbol{I}_N \\ & & -\boldsymbol{I}_N & 2\boldsymbol{I}_N \end{bmatrix}$$

因此，(6.17)式与(6.16)[1]式相符合.

为计算由克罗内克积定义的矩阵 $\boldsymbol{T}_{N\times N}$ 的特征值，需要下列引理，其证明也是问题 6.4 的一部分.

引理 6.3　下列关于克罗内克积的事实成立：

275

1. 假定积 $\boldsymbol{A}\cdot\boldsymbol{C}$ 和 $\boldsymbol{B}\cdot\boldsymbol{D}$ 完全确定. 则 $(\boldsymbol{A}\otimes\boldsymbol{B})\cdot(\boldsymbol{C}\otimes\boldsymbol{D}) = (\boldsymbol{A}\cdot\boldsymbol{C})\otimes(\boldsymbol{B}\cdot\boldsymbol{D})$.
2. 若 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 可逆，则 $(\boldsymbol{A}\otimes\boldsymbol{B})^{-1} = \boldsymbol{A}^{-1}\otimes\boldsymbol{B}^{-1}$.
3. $(\boldsymbol{A}\otimes\boldsymbol{B})^{\mathrm{T}} = \boldsymbol{A}^{\mathrm{T}}\otimes\boldsymbol{B}^{\mathrm{T}}$.

命题 6.1　设 $\boldsymbol{T}_N = \boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{Z}^{\mathrm{T}}$ 是 $\boldsymbol{T}_N$ 的特征分解，$\boldsymbol{Z} = [z_1,\cdots,z_N]$ 是正交阵，其列是特征向量，而 $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1,\cdots,\lambda_N)$. 则 $\boldsymbol{T}_{N\times N} = \boldsymbol{I}\otimes\boldsymbol{T}_N + \boldsymbol{T}_N\otimes\boldsymbol{I}$ 的特征分解是

$$\boldsymbol{I}\otimes\boldsymbol{T}_N + \boldsymbol{T}_N\otimes\boldsymbol{I} = (\boldsymbol{Z}\otimes\boldsymbol{Z})\cdot(\boldsymbol{I}\otimes\boldsymbol{\Lambda} + \boldsymbol{\Lambda}\otimes\boldsymbol{I})\cdot(\boldsymbol{Z}\otimes\boldsymbol{Z})^{\mathrm{T}}. \tag{6.18}$$

$\boldsymbol{I}\otimes\boldsymbol{\Lambda} + \boldsymbol{\Lambda}\otimes\boldsymbol{I}$ 是对角阵，其第 $(iN+j)$ 个对角元是 $\boldsymbol{T}_{N\times N}$ 的第 (i, j) 个特征值 $\lambda_{ij} = \lambda_i + \lambda_j$. $\boldsymbol{Z}\otimes\boldsymbol{Z}$ 是正交阵，其第 $(iN+j)$ 列对应的特征向量是 $z_i\otimes z_j$.

证明　从引理 6.3 的第 1 和第 3 部分知，容易验证 $\boldsymbol{Z}\otimes\boldsymbol{Z}$ 是正交的，因为 $(\boldsymbol{Z}\otimes\boldsymbol{Z})(\boldsymbol{Z}\otimes\boldsymbol{Z})^{\mathrm{T}} = (\boldsymbol{Z}\otimes\boldsymbol{Z})(\boldsymbol{Z}^{\mathrm{T}}\otimes\boldsymbol{Z}^{\mathrm{T}}) = (\boldsymbol{Z}\cdot\boldsymbol{Z}^{\mathrm{T}})\otimes(\boldsymbol{Z}\cdot\boldsymbol{Z}^{\mathrm{T}}) = \boldsymbol{I}\otimes\boldsymbol{I} = \boldsymbol{I}$. 现在可验证(6.18)：

$$\begin{aligned}
&(\boldsymbol{Z}\otimes\boldsymbol{Z})\cdot(\boldsymbol{I}\otimes\boldsymbol{\Lambda} + \boldsymbol{\Lambda}\otimes\boldsymbol{I})\cdot(\boldsymbol{Z}\otimes\boldsymbol{Z})^{\mathrm{T}} \\
&= (\boldsymbol{Z}\otimes\boldsymbol{Z})\cdot(\boldsymbol{I}\otimes\boldsymbol{\Lambda} + \boldsymbol{\Lambda}\otimes\boldsymbol{I})\cdot(\boldsymbol{Z}^{\mathrm{T}}\otimes\boldsymbol{Z}^{\mathrm{T}}) \quad \text{由引理 6.3 的第 3 部分} \\
&= (\boldsymbol{Z}\cdot\boldsymbol{I}\cdot\boldsymbol{Z}^{\mathrm{T}})\otimes(\boldsymbol{Z}\cdot\boldsymbol{\Lambda}\cdot\boldsymbol{Z}^{\mathrm{T}}) + (\boldsymbol{Z}\cdot\boldsymbol{\Lambda}\cdot\boldsymbol{Z}^{\mathrm{T}})\otimes(\boldsymbol{Z}\cdot\boldsymbol{I}\cdot\boldsymbol{Z}^{\mathrm{T}}) \quad \text{由引理 6.3 的第 1 部分} \\
&= (\boldsymbol{I})\otimes(\boldsymbol{T}_N) + (\boldsymbol{T}_N)\otimes(\boldsymbol{I}) \\
&= \boldsymbol{T}_{N\times N}.
\end{aligned}$$

也容易验证 $\boldsymbol{I}\otimes\boldsymbol{\Lambda} + \boldsymbol{\Lambda}\otimes\boldsymbol{I}$ 是对角阵，第 $(iN+j)$ 个对角元为 $\lambda_j + \lambda_i$，所以(6.18)式确实是 $\boldsymbol{T}_{N\times N}$ 的特征分解. 最后，从克罗内克积的定义可以看出 $\boldsymbol{Z}\otimes\boldsymbol{Z}$ 的第 $iN+j$ 列是 $z_i\otimes z_j$. □

---

1. 我们可以用这个公式以两行 Matlab 计算 $\boldsymbol{T}_{N\times N}$

```
TN = 2 为 eye(N) - diag(ones(N - 1,1),1) - diag(ones(N - 1,1), - 1);
TN × N = kron(eye(N),TN) + kron(TN,eye(N));
```

读者可以证实特征向量 $z_i \otimes z_j = \text{vec}(z_j z_i^T)$，从而与(6.12)式中特征向量的表达式一致.

把命题6.1推广到求解西尔维斯特方程 $\boldsymbol{AX}-\boldsymbol{XB}=\boldsymbol{C}$ 时产生的矩阵 $\boldsymbol{A}\otimes\boldsymbol{I}+\boldsymbol{B}^T\otimes\boldsymbol{I}$，见问题6.5(以及问题4.6).

类似地，三维泊松方程导致

$$\boldsymbol{T}_{N\times N\times N}\equiv\boldsymbol{T}_N\otimes\boldsymbol{I}_N\otimes\boldsymbol{I}_N+\boldsymbol{I}_N\otimes\boldsymbol{T}_N\otimes\boldsymbol{I}_N+\boldsymbol{I}_N\otimes\boldsymbol{I}_N\otimes\boldsymbol{T}_N,$$

它的特征值是由 $\boldsymbol{T}_N$ 的特征值的所有可能的三元序组之和构成而特征向量矩阵是 $\boldsymbol{Z}\otimes\boldsymbol{Z}\otimes\boldsymbol{Z}$. 高维的泊松方程可类似地表示.

276

表6-1 在 $N\times N$ 网格上求解泊松方程的复杂性的阶($n=N^2$)

方 法	串行时间	空间	直接或迭代法	节
稠密楚列斯基	n^3	n^2	直接	2.7.1
显式求逆	n^2	n^2	直接	
带状楚列斯基	n^2	$n^{3/2}$	直接	2.7.3
雅可比	n^2	n	迭代	6.5
高斯-塞德尔	n^2	n	迭代	6.5
稀疏楚列斯基	$n^{3/2}$	$n\cdot\log n$	直接	2.7.4
共轭梯度	$n^{3/2}$	n	迭代	6.6
逐次超松弛	$n^{3/2}$	n	迭代	6.5
带切比雪夫加速的SSOR	$n^{5/4}$	n	迭代	6.5
快速傅里叶变换	$n\cdot\log n$	n	直接	6.7
块循环约化	$n\cdot\log n$	n	直接	6.8
多重网格	n	n	迭代	6.9
下 界	n	n		

6.4 解泊松方程方法小结

表6-1列举在 $N\times N$ 网格上求解模型问题的各种直接法和迭代法的代价. 变量 $n=N^2$ 是未知量个数. 因为直接法提供精确解(不计舍入时)，而迭代法只提供近似解，所以当比较它们的代价时必须小心. 因为可用迭代法更便宜地算出一个低精度的解而不是高精度的解. 所以，假定迭代法通常作足够次迭代使误差至多是某个固定的小量[1](譬如，10^{-6})时，我们比较代价.

表6-1的第二列和第三列给出在一台串行机上所需的算术运算次数(或时间)和空间数. 第四列指出方法是直接的还是迭代的. 所有的表列数据都用 $O(\cdot)$ 理解；常数与执行过程细节以及迭代法的停止准则(譬如说，10^{-6})有关. 例如，关于楚列斯基的表列数据也应用于高斯消去法，因为这仅仅改变常数为2的因子. 最后列指

1. 另一方面，我们可以迭代直到误差为截断误差大小 $O(h^2)=O((N+1)^{-2})$，并且可以证明这样做将使表6-1中迭代法的代价增加 $O(\log n)$ 的倍数.

出算法在课本中讨论的地方.

277 方法是按速度递增次序列举的，从最慢的(稠密楚列斯基)到最快的(多重网格)，最后以应用于任何方法的一个下界结尾. 因为每个解分量至少需要一次运算，所以下界为 n，否则它们不能都不同并且也依赖于输入. 粗略地讲，方法也是按普遍适用性递减次序列举的，稠密的楚列斯基可适用于任何对称正定阵而后面的算法仅对有限的矩阵类可适用(或者至少可证明收敛). 在后面几节中将更详尽地描述各种各样方法的适用性.

"显式求逆"算法涉及预先计算 $T_{N\times N}$ 的显式求逆，并用单独的矩阵-向量乘法计算 $v=T_{N\times N}^{-1}f$(不计预先计算 $T_{N\times N}^{-1}$ 的 flops). 连同稠密的楚列斯基一起，它共使用 n^2 个空间，大大地多于其他方法. 它不是一个好的方法. 在 2.7.3 节中讨论了带状楚列斯基方法，这个方法利用了 $2N+1$ 条对角线带外面没有元素计算或存放的优势.

雅可比和高斯塞德尔是经典的迭代法，虽然不是特别快速的，但它们可构成其他一些较快算法：逐次超松弛法，对称逐次超松弛法及最快的算法多重网格法的基础. 故将在 6.5 节中更详尽地研究它们.

稀疏楚列斯基涉及 2.7.4 节中讨论的算法：它是楚列斯基算法的一种执行过程，它避免存放或运算 $T_{N\times N}$ 的零元素或者它的楚列斯基因子. 而且为了极小化工作量和存储量，我们假定 $T_{N\times N}$ 的行和列已被"优化排序"(利用嵌套剖分[112, 113]). 虽然稀疏楚列斯基对二维泊松方程是相当快的，但是它在三维情况是非常坏的(利用 $O(N^6)=O(n^2)$ 时间和 $O(N^4)=O(n^{4/3})$ 空间)，因为在算法过程间有许多零元素"填补".

共轭梯度法是称为*克雷洛夫子空间*(krylov subspace)方法的一大类方法代表，这类方法极其广泛地应用于解线性方程组和求稀疏矩阵的特征值. 在 6.6 节中将更详尽地讨论这些方法.

最快的方法是块循环约化，快速傅里叶变换(FFT)以及多重网格法. 特别地，多重网格法对每个解分量只做 $O(1)$ 次运算，这个方法是渐近最优的.

最后的提醒是这个表格并未给出一个彻底的描述，因为并未标明常数. 对于在一台特别的机器上的一个特殊大小的问题，人们不能直接推断何种方法最快. 对足够大的 n，尽管像雅可比法、高斯-塞德尔法，共轭梯度法和逐次超松弛法那样的迭代法虽然次于 FFT，块循环约化和多重网格法，但是因为它们构造一些较快的方法的块并且还比那些较快的方法应用于更大类的问题，所以它们继续保持重要性.

278 所有这些算法可以并行地执行；细节见 PARALLEL_HOMEPAGE 上的讲义. 有趣的是依赖于并行机，多重网格法可能不再是最快的算法. 这是因为在一台并行机上把不相连的处理机数据交换到另一台处理机所需的时间可能像浮点运算一样昂贵，而其他的算法可能交换少于多重网格法.

6.5 基本迭代法

本节中将讨论大部分基本迭代法：

雅可比；

高斯 - 塞德尔；

逐次超松弛（SOR(ω)）；

带对称超松弛的切比雪夫加速（SSOR(ω)）.

在 NETLIB/templates 上也讨论这些方法并且提供它们的执行过程.

给定 x_0，这些方法生成一个收敛于 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解 $\boldsymbol{A}^{-1}\boldsymbol{b}$ 的序列 x_m，其中 x_{m+1} 能从 x_m 方便地算出.

定义 6.3 $\boldsymbol{A}$ 的分裂是一个分解 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{K}$，其中 $\boldsymbol{M}$ 非奇异.

一个分裂得到一个迭代法如下：$\boldsymbol{Ax}=\boldsymbol{Mx}-\boldsymbol{Kx}=\boldsymbol{b}$ 推出 $\boldsymbol{Mx}=\boldsymbol{Kx}+\boldsymbol{b}$ 或 $\boldsymbol{x}=\boldsymbol{M}^{-1}\boldsymbol{Kx}+\boldsymbol{M}^{-1}\boldsymbol{b}\equiv\boldsymbol{Rx}+\boldsymbol{c}$. 故可取 $\boldsymbol{x}_{m+1}=\boldsymbol{Rx}_m+\boldsymbol{c}$ 作为迭代法. 让我们观察它何时收敛.

引理 6.4 设 $\|\cdot\|$ 是任意的算子范数 $\left(\|\boldsymbol{R}\|\equiv\max_{x\neq0}\dfrac{\|\boldsymbol{Rx}\|}{\|\boldsymbol{x}\|}\right)$. 若 $\|\boldsymbol{R}\|<1$，则 $\boldsymbol{x}_{m+1}=\boldsymbol{Rx}_m+\boldsymbol{c}$ 对任意 $\boldsymbol{x}_0$ 收敛.

证明 从 $\boldsymbol{x}_{m+1}=\boldsymbol{Rx}_m+\boldsymbol{c}$ 中减去 $\boldsymbol{x}=\boldsymbol{Rx}+\boldsymbol{c}$ 得到 $\boldsymbol{x}_{m+1}-\boldsymbol{x}=\boldsymbol{R}(\boldsymbol{x}_m-\boldsymbol{x})$. 于是 $\|\boldsymbol{x}_{m+1}-\boldsymbol{x}\|\leqslant\|\boldsymbol{R}\|\cdot\|\boldsymbol{x}_m-\boldsymbol{x}\|\leqslant\|\boldsymbol{R}\|^{m+1}\cdot\|\boldsymbol{x}_0-\boldsymbol{x}\|$，因为 $\|\boldsymbol{R}\|<1$，所以它收敛于 0. □

最终的收敛准则将依赖于下列 $\boldsymbol{R}$ 的性质.

定义 6.4 $\boldsymbol{R}$ 的谱半径是 $\rho(\boldsymbol{R})\equiv\max|\lambda|$，其中最大值是取遍 $\boldsymbol{R}$ 的所有的特征值 λ.

引理 6.5 对一切算子范数 $\rho(\boldsymbol{R})\leqslant\|\boldsymbol{R}\|$. 对一切 $\boldsymbol{R}$ 和所有 $\varepsilon>0$ 存在一个算子范数 $\|\cdot\|_\star$ 使得 $\|\boldsymbol{R}\|_\star\leqslant\rho(\boldsymbol{R})+\varepsilon$. 范数 $\|\cdot\|_\star$ 同时依赖于 $\boldsymbol{R}$ 和 ε.

证明 证明对任意的算子范数 $\rho(\boldsymbol{R})\leqslant\|\boldsymbol{R}\|$，设 $\boldsymbol{x}$ 是关于 λ 的特征向量，其中 $\rho(\boldsymbol{R})=|\lambda|$，故 $\|\boldsymbol{R}\|=\max_{y\neq0}\dfrac{\|\boldsymbol{Ry}\|}{\|\boldsymbol{y}\|}\geqslant\dfrac{\|\boldsymbol{Rx}\|}{\|\boldsymbol{x}\|}=\dfrac{\|\lambda\boldsymbol{x}\|}{\|\boldsymbol{x}\|}=|\lambda|$.

279

构造一个算子范数 $\|\cdot\|_\star$ 使得 $\|\boldsymbol{R}\|_\star\leqslant\rho(\boldsymbol{R})+\varepsilon$，设 $\boldsymbol{S}^{-1}\boldsymbol{RS}=\boldsymbol{J}$ 是若尔当型. 设 $\boldsymbol{D}_\varepsilon=\mathrm{diag}(1,\varepsilon,\varepsilon^2,\cdots,\varepsilon^{n-1})$. 则

$$
\begin{aligned}
(\boldsymbol{SD}_\varepsilon)^{-1}\boldsymbol{R}(\boldsymbol{SD}_\varepsilon)&=\boldsymbol{D}_\varepsilon^{-1}\boldsymbol{JD}_\varepsilon\\
&=\left[\begin{array}{c|c|c}
\begin{matrix}\lambda_1 & \varepsilon & & \\ & \ddots & \ddots & \\ & & \ddots & \varepsilon \\ & & & \lambda_1\end{matrix} & & \\ \hline
 & \begin{matrix}\lambda_2 & \varepsilon & & \\ & \ddots & \ddots & \\ & & \ddots & \varepsilon \\ & & & \lambda_2\end{matrix} & \\ \hline
 & & \ddots
\end{array}\right],
\end{aligned}
$$

即这是一个在对角线上面有 ε 的“若尔当型”. 利用向量范数 $\|\boldsymbol{x}\|_\star\equiv\|(\boldsymbol{SD}_\varepsilon)^{-1}\boldsymbol{x}\|_\infty$ 生成算子范数

$$\|\boldsymbol{R}\|_\star\equiv\max_{x\neq0}\frac{\|\boldsymbol{Rx}\|_\star}{\|\boldsymbol{x}\|_\star}$$

$$= \max_{x \neq 0} \frac{\|(SD_\varepsilon)^{-1} Rx\|_\infty}{\|(SD_\varepsilon)^{-1} x\|_\infty}$$

$$= \max_{y \neq 0} \frac{\|(SD_\varepsilon)^{-1} R(SD_\varepsilon) y\|_\infty}{\|y\|_\infty}$$

$$= \|(SD_\varepsilon)^{-1} R(SD_\varepsilon)\|_\infty$$

$$= \max_i |\lambda_i| + \varepsilon$$

$$= \rho(R) + \varepsilon.$$ □

定理 6.1　迭代 $x_{m+1} = Rx_m + c$ 对所有的初始向量 x_0 和所有的 b 收敛于 $Ax = b$ 的解当且仅当 $\rho(R) < 1$.

证明　若 $\rho(R) \geq 1$，选择 $x_0 - x$ 是关于特征值 λ 的一个特征向量，其中 $|\lambda| = \rho(R)$. 则

$$(x_{m+1} - x) = R(x_m - x) = \cdots = R^{m+1}(x_0 - x) = \lambda^{m+1}(x_0 - x)$$

将不趋于0. 若 $\rho(R) < 1$，利用引理6.5选择一个算子范数使得 $\|R\|_\star < 1$，然后应用引理6.4得出方法收敛. □

280

定义 6.5　$x_{m+1} = Rx_m + c$ 收敛速率是 $r(R) \equiv -\log_{10}\rho(R)$.

$r(R)$ 是每次迭代在解中正确的小数位个数的增量，因为 $\log_{10}\|x_m - x\|_\star - \log_{10}\|x_{m+1} - x\|_\star \geq r(R) + O(\varepsilon)$. $\rho(R)$ 越小，收敛速率越高，即每次迭代计算的正确小数位数越多.

现在我们的目标是选择一个分裂 $A = M - K$ 同时满足

(1) $Rx = M^{-1}Kx$ 和 $c = M^{-1}b$ 都容易计算.

(2) $\rho(R)$ 是小的.

我们将需要平衡这些相矛盾的目标. 例如，选择 $M = I$ 对目标(1)是好的，但可能不产生 $\rho(R) < 1$. 另一方面，选择 $M = A$ 和 $K = 0$ 对目标(2)是好的，但对目标(1)可能是坏的.

本节中所讨论的方法的分裂全部共用下列记号. 当 A 在它的对角线上无零元时，记

$$A = D - \tilde{L} - \tilde{U} = D(I - L - U), \tag{6.19}$$

其中 D 是 A 的对角线部分，$-\tilde{L}$ 是 A 的严格下三角部分，$DL = \tilde{L}$，$-\tilde{U}$ 是 A 的严格上三角部分，$DU = \tilde{U}$.

6.5.1　雅可比法

雅可比法可描述为通过方程反复地作循环，改变变量 j 以使第 j 个方程精确地成立. 利用(6.19)式的记号，雅可比法的分裂是 $A = D - (\tilde{L} + \tilde{U})$. 记 $R_J \equiv D^{-1}(\tilde{L} + \tilde{U}) = L + U$ 和 $c_J \equiv D^{-1}b$，因此可把雅可比法的单步写成 $x_{m+1} = R_J x_m + c_J$. 为看出这个公式对应于雅可比法的第一次描述，注意公式蕴涵 $Dx_{m+1} = (\tilde{L} + \tilde{U})x_m + b$，或 $a_{jj}x_{m+1,j} = -\sum_{k \neq j} a_{jk}x_{m,k} + b_j$，或 $a_{jj}x_{m+1,j} + \sum_{k \neq j} a_{jk}x_{m,k} = b_j$.

算法6.1 雅可比法的单步

for $j = 1$ to n

$$x_{m+1,j} = \frac{1}{a_{jj}}\left(b_j - \sum_{k \neq j} a_{jk} x_{m,k}\right)$$

end for

在模型问题的特殊情况中，雅可比法的执行过程简化如下. 直接从方程(6.10)操作并设 $v_{m,i,j}$ 表示在网格点 i，j 上解的第 m 个值，雅可比法变成下列

算法6.2 二维泊松方程雅可比法的单步：

for $i = 1$ to N

  for $j = 1$ to N

$$v_{m+1,i,j} = (v_{m,i-1,j} + v_{m,i+1,j} + v_{m,i,j-1} + v_{m,i,j+1} + h^2 f_{ij})/4$$

  end for

end for

281

换言之，在每步通过“平均”v_{ij} 的相邻值和 $h^2 f_{ij}$ 来得到 v_{ij} 的新值. 注意所有的新值 $v_{m+1,i,j}$ 可以彼此独立地计算. 实际上，当 $v_{m+1,i,j}$ 被存放在一个正方形数组 $\hat{V}$ 中，这个数组包含额外的第一行和最后一行零元以及的第一列和最后一列零元时，算法6.2可在 Matlab 的一个程序行中执行(见问题6.6).

6.5.2 高斯-塞德尔法

这个方法的促动因素是在雅可比法循环的第 j 步，已经具有解的前 $j-1$ 个分量的改进值，因而我们在和式中应该利用这些值.

算法6.3 高斯-塞德尔法的单步：

for $j = 1$ to n

$$x_{m+1,j} = \frac{1}{a_{jj}}\Big(b_j - \underbrace{\sum_{k=1}^{j-1} a_{jk} x_{m+1,k}}_{\text{更新的}x} - \underbrace{\sum_{k=j+1}^{n} a_{jk} x_{m,k}}_{\text{原来的}x}\Big)$$

end for

为了后面分析之用，需要把这个算法写成如下形式 $\boldsymbol{x}_{m+1} = \boldsymbol{R}_{GS}\boldsymbol{x}_m + \boldsymbol{c}_{GS}$. 为此，注意它首先可改写为

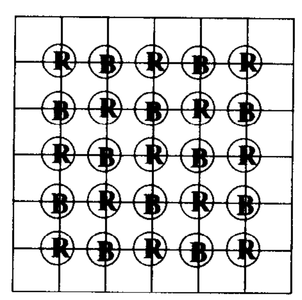
$$\sum_{k=1}^{j} a_{jk} x_{m+1,k} = -\sum_{k=j+1}^{n} a_{jk} x_{m,k} + b_j. \tag{6.20}$$

然后利用(6.19)式的记号，把(6.20)式改写为$(D-\widetilde{L})x_{m+1}=\widetilde{U}x_m+b$或

$$
\begin{aligned}
x_{m+1} &= (D-\widetilde{L})^{-1}\widetilde{U}x_m+(D-\widetilde{L})^{-1}b \\
&= (I-L)^{-1}Ux_m+(I-L)^{-1}D^{-1}b\equiv R_{GS}x_m+c_{GS}.
\end{aligned}
$$

正如与雅可比法一样，考虑如何对模型问题实施高斯-塞德尔法. 除了必须遵守哪些变量是新的(编号为$m+1$)和哪些是原来的(编号为m)的方针外，原则上还是十分相似的. 但是依赖于经由网格点i, j循环的次序，将得到高斯-塞德尔法的不同的(且正确的)执行过程. 这不像雅可比法，在那里我们更新变量的次序是不相关的. 例如，若我们(在任何$v_{m,i,j}$之前)首先更新$v_{m,1,1}$，则它的所有相邻的值必定是老的. 但是若我们最后更新$v_{m,1,1}$，则它的所有相邻的值必定是新的，因而得到$v_{m,1,1}$不同的值. 事实上，有与N^2个变量阶数一样多的许多可能的方式(即$N^2!$)执行高斯-塞德尔法. 但是在所有这些次序中的两个次序最为重要. 第一个是图6-4中指出的次序，这个称为自然次序(natural ordering).

第二个次序称为红-黑次序(red-black ordering). 这是重要的次序，因为在6.5.4节和6.5.5节中的最佳收敛性结果依赖于它. 为说明红黑次序，考虑下面类似像棋盘着色的未知量网格的图形. Ⓑ节点对应棋盘上的黑方格，而Ⓡ节点对应于红方格(Ⓑ表示黑节点，Ⓡ表示红节点).



红-黑次序是按照红节点在黑节点之前排序. 注意红节点只有黑节点是相邻的. 故当我们首先更新所有的红节点时，它们只使用来自黑节点的老的数据. 然后，当更新只是相邻于红节点的黑节点时，它们将只使用来自红节点的新数据. 于是可得下列算法.

算法6.4 对具有红-黑次序的二维泊松方程的高斯-塞德尔方法的单步：

for 所有的(Ⓡ)节点 i, j

$$v_{m+1,i,j}=(v_{m,i-1,j}+v_{m,i+1,j}+v_{m,i,j-1}+v_{m,i,j+1}+h^2f_{ij})/4$$

end for

for 所有的Ⓑ节点 i, j

$$v_{m+1,i,j}=(v_{m+1,i-1,j}+v_{m+1,i+1,j}+v_{m+1,i,j-1}+v_{m+1,i,j+1}+h^2f_{ij})/4$$

end for

6.5.3 逐次超松弛法

我们称这个方法为 SOR(ω)，其中 ω 为松弛参数(relaxation parameter). 促动因素是取 $x_{m+1,j}$和 $x_{m,j}$的一个适当的加权平均来改进高斯-高德尔循环： 283

$$\text{SOR 的 } x_{m+1,j} = (1-\omega)x_{m,j} + \omega x_{m+1,j}$$

得到下列算法.

算法 6.5 SOR:

for $j=1$ to n

$$x_{m+1,j} = (1-\omega)x_{m,j} + \frac{\omega}{a_{jj}}\left[b_j - \sum_{k=1}^{j-1} a_{jk}x_{m+1,k} - \sum_{k=j+1}^{n} a_{jk}x_{m,k}\right]$$

end for

可以重新安排这个计算得到，对 $j=1$ 到 n，

$$a_{jj}x_{m+1,j} + \omega\sum_{k=1}^{j-1} a_{jk}x_{m+1,k} = (1-\omega)a_{jj}x_{m,j} - \omega\sum_{k=j+1}^{n} a_{jk}x_{m,k} + \omega b_j$$

或者再利用(6.19)式的记号，得到

$$(\boldsymbol{D} - \omega\tilde{\boldsymbol{L}})\boldsymbol{x}_{m+1} = ((1-\omega)\boldsymbol{D} + \omega\tilde{\boldsymbol{U}})\boldsymbol{x}_m + \omega\boldsymbol{b}$$

或者

$$\begin{aligned}\boldsymbol{x}_{m+1} &= (\boldsymbol{D} - \omega\tilde{\boldsymbol{L}})^{-1}((1-\omega)\boldsymbol{D} + \omega\tilde{\boldsymbol{U}})x_m + \omega(\boldsymbol{D} - \omega\tilde{\boldsymbol{L}})^{-1}\boldsymbol{b} \\ &= (\boldsymbol{I} - \omega\boldsymbol{L})^{-1}((1-\omega)\boldsymbol{I} + \omega\boldsymbol{U})x_m + \omega(\boldsymbol{I} - \omega\boldsymbol{L})^{-1}\boldsymbol{D}^{-1}\boldsymbol{b} \\ &\equiv \boldsymbol{R}_{\text{SOR}(\omega)}x_m + \boldsymbol{c}_{\text{SOR}(\omega)}.\end{aligned} \tag{6.21}$$

根据 ω 的值区分三种情况：$\omega=1$ 等价于高斯-塞德尔法，$\omega<1$ 称为低松弛(underrelaxation)，而 $\omega>1$ 称为超松弛(overrelaxation). 关于超松弛的一种有点表面的促动因素是若从 x_m 到 x_{m+1}的方向是移动解的一个好方向，则在那个方向上移动超过 $\omega>1$ 倍更好.

在下面的两节中，将指出如何对模型问题取最佳的 ω. 这个最佳性依赖于使用红黑次序.

算法 6.6 具有红-黑次序的二维泊松方程的 SOR(ω)的单步：

for 所有的(Ⓡ)节点 i, j

$$v_{m+1,i,j} = (1-\omega)v_{m,i,j} + \omega(v_{m,i-1,j} + v_{m,i+1,j} + v_{m,i,j-1} + v_{m,i,j+1} + h^2 f_{ij})/4$$

end for

for 所有的(Ⓑ)节点 i,j

$$v_{m+1,i,j} = (1-\omega)v_{m,i,j} + \omega(v_{m+1,i-1,j} + v_{m+1,i+1,j} + v_{m+1,i,j-1} + v_{m+1,i,j+1} + h^2 f_{ij})/4$$

end for

284

6.5.4 模型问题的雅可比、高斯－塞德尔和SOR(ω)法的收敛性

对模型问题容易计算雅可比法收敛多少快，因为对应的分裂是 $\boldsymbol{T}_{N\times N}=4\boldsymbol{I}-(4\boldsymbol{I}-\boldsymbol{T}_{N\times N})$，故 $\boldsymbol{R}_J=(4\boldsymbol{I})^{-1}(4\boldsymbol{I}-\boldsymbol{T}_{N\times N})=\boldsymbol{I}-\boldsymbol{T}_{N\times N}/4$. 于是 $\boldsymbol{R}_J$ 的特征值是 $1-\lambda_{i,j}/4$，其中 $\lambda_{i,j}$ 是 $\boldsymbol{T}_{N\times N}$ 的特征值：

$$\lambda_{i,j}=\lambda_i+\lambda_j=4-2\left(\cos\frac{\pi i}{N+1}+\cos\frac{\pi j}{N+1}\right).$$

$\rho(\boldsymbol{R}_J)$ 是 $|1-\lambda_{i,j}/4|$ 的最大值，即

$$\rho(\boldsymbol{R}_J)=|1-\lambda_{1,1}/4|=|1-\lambda_{N,N}/4|=\cos\frac{\pi}{N+1}\approx 1-\frac{\pi^2}{2(N+1)^2}.$$

注意当 N 增长且 $\boldsymbol{T}$ 变成更病态时，谱半径 $\rho(\boldsymbol{R}_J)$ 逼近1. 因为误差在每一步用谱半径相乘，所以收敛慢下来. 为更精确地估计收敛速度，我们用 $\mathrm{e}^{-1}=\exp(-1)$ 计算减少误差所需的雅可比迭代的次数 m. 于是 m 必须满足 $(\rho(R_J))^m=\mathrm{e}^{-1}$ 或 $\left(1-\frac{\pi^2}{2(N+1)^2}\right)^m=\mathrm{e}^{-1}$，或 $m\approx\frac{2(N+1)^2}{\pi^2}=O(N^2)=O(n)$. 因而迭代次数与未知量个数成比例. 因为雅可比的单步花费 $O(1)$ 去更新每个解分量或者花费 $O(n)$ 去更新它们全部，所以用 e^{-1}（或用任何小于1的因子）减少误差它花费 $O(n^2)$. 这就解释了表6-1中关于雅可比法的表列数据.

这是一个普遍的现象：原问题越病态，大部分迭代法的收敛速度越慢. 有重要的例外，例如在后面讨论的多重网格法和区域分解法.

在下节中将指出，倘若泊松方程中的变量以红－黑次序更新（见算法6.4和推论6.1），则 $\rho(\boldsymbol{R}_{GS})=\rho(\boldsymbol{R}_J)^2=\cos^2\frac{\pi}{N+1}$. 换言之，单步高斯－塞德尔减少误差像二步雅可比法那样多. 这对某些用有限差分近似逼近微分方程所产生的矩阵是一个普遍的现象. 这也说明了表6-1中关于高斯－塞德尔法的表列数据，因为它的速度只是像雅可比法的二倍那样快，所以在 $O(\cdot)$ 意义下它仍旧有相同的复杂性.

对相同的红－黑更新次序（见算法6.6和定理6.7），关于松弛因子 $1<\omega=2/\left(1+\sin\frac{\pi}{N+1}\right)<2$，也将证明

$$\rho(\boldsymbol{R}_{\mathrm{SOR}(\omega)})=\frac{\cos^2\frac{\pi}{N+1}}{\left(1+\sin\frac{\pi}{N+1}\right)^2}\approx 1-\frac{2\pi}{N+1},\text{对大的 } N.$$

285　这与 R_J 和 R_{GS} 的 $\rho(R)=1-O\left(\frac{1}{N^2}\right)$ 大不相同. 这是 ω 的最佳值，即它极小化 $R_{\mathrm{SOR}(\omega)}$. 对这样选择的 ω，SOR(ω)近似地比雅可比法或高斯－塞德尔法快 N 倍，因为若SOR(ω)作 j 步减少误差与雅可比法或高斯－塞德尔法作 k 步一样多，则 $\left(1-\frac{1}{N^2}\right)^k\approx$

$\left(1-\frac{1}{N}\right)^j$，推出 $1-\frac{k}{N^2}\approx 1-\frac{j}{N}$ 或 $k\approx j\cdot N$. 正如表 6-1 中所指出的那样，这就把 SOR(ω)的复杂性从 $O(n^2)$ 降到 $O(n^{3/2})$.

下节中将指出对某些有限差分矩阵一般如何选择 ω 去极小化 $\rho(R_{\text{SOR}(\omega)})$.

6.5.5 雅可比、高斯-塞德尔和 SOR(ω)法明细的收敛准则

我们将给出保证这些方法收敛的一系列准则. 第一个准则计算是简单的但并非总是能应用的，特别地，不能用于模型问题. 然后我们给出几个更复杂的准则，它对矩阵 $\boldsymbol{A}$ 提出较强的条件. 但是作为报答给出更多有关收敛的信息. 这些更复杂的准则特别适合离散化像泊松方程那样的一类偏微分方程产生的矩阵.

下面是本节结论的概要：

1. 若 $\boldsymbol{A}$ 是严格行对角占优矩阵(定义 6.6)，则雅可比法和高斯-塞德尔法都收敛，并且高斯-塞德尔法较快(定理 6.2). 严格行对角占优意味着 $\boldsymbol{A}$ 的每个对角元数量上大于该行中其他元素数量之和.

2. 因为模型问题不是严格行对角占优，所以上述结果不能应用. 因而我们寻找一个较弱的对角占优形式(定义 6.11). 为证明雅可比法和高斯-塞德尔法收敛对 $\boldsymbol{A}$ 的非零元的模式强加一个所谓不可约性(irreducibility)条件(定义 6.7). 高斯-塞德尔法收敛再次快于雅可比法(定理 6.3). 这个结论应用于模型问题.

3. 转到 SOR(ω)，证明 $0<\omega<2$ 对收敛是必要的(定理 6.4). 若 $\boldsymbol{A}$ 又是正定的(如同模型问题)，则 $0<\omega<2$ 对收敛也是充分的(定理 6.5).

4. 为了数量上比较雅可比法、高斯-塞德尔法和 SOR(ω)法，我们关于 $\boldsymbol{A}$ 的非零元的模式作出更多的假设. 这个性质称为性质 A(定义 6.12). 这个概念相当于说矩阵的图是分为两部分的(graph of the matrix is bipartite). 性质 A 本质上表示可以利用红-黑次序更新变量. 给定性质 A，存在 R_J，R_{GS} 和 $R_{\text{SOR}(\omega)}$ 之特征值相关的一个简单的代数公式(定理 6.6)，由此可比较这些方法的收敛速率. 这个公式也可计算使得 SOR(ω)收敛尽可能快的最佳 ω(定理 6.7).

286

定义 6.6 若对一切 i，$|a_{ii}|>\sum_{j\neq i}|a_{ij}|$，则 $\boldsymbol{A}$ 是严格行对角占优的.

定理 6.2 若 $\boldsymbol{A}$ 严格行对角占优，则雅可比法和高斯-塞德尔法都收敛. 事实上 $\|\boldsymbol{R}_{GS}\|_\infty\leq\|\boldsymbol{R}_J\|_\infty<1$.

不等式 $\|R_{GS}\|_\infty\leq\|R_J\|_\infty$ 推出对最差的问题关于高斯-塞德尔法的单步收敛至少快于最差的问题关于雅可比法的单步. 它并不保证对任何特别的 $\boldsymbol{Ax}=\boldsymbol{b}$，高斯-塞德尔法应该比雅可比法更快；雅可比法可能"偶然地"在某步有较小的误差.

证明 再次利用(6.19)式的记号，记 $\boldsymbol{R}_J=\boldsymbol{L}+\boldsymbol{U}$ 和 $\boldsymbol{R}_{GS}=(\boldsymbol{I}-\boldsymbol{L})^{-1}\boldsymbol{U}$. 我们要证明

$$\|\boldsymbol{R}_{GS}\|_\infty=\|\,|\boldsymbol{R}_{GS}|\boldsymbol{e}\|_\infty\leq\|\,|\boldsymbol{R}_J|\boldsymbol{e}\|_\infty=\|\boldsymbol{R}_J\|_\infty, \tag{6.22}$$

其中 $\boldsymbol{e}=[1,\cdots,1]^{\mathrm{T}}$ 是全 1 向量. 若能证明较强的按分量的不等式

$$|(\boldsymbol{I}-\boldsymbol{L})^{-1}\boldsymbol{U}|\cdot\boldsymbol{e}=|\boldsymbol{R}_{GS}|\cdot\boldsymbol{e}\leq|\boldsymbol{R}_J|\cdot\boldsymbol{e}=(|\boldsymbol{L}|+|\boldsymbol{U}|)\cdot\boldsymbol{e} \tag{6.23}$$

则不等式(6.22)成立. 因为

$$
\begin{aligned}
|(I-L)^{-1}U|\cdot e &\leqslant |(I-L)^{-1}|\cdot|U|\cdot e && \text{由三角不等式}\\
&= \left|\sum_{i=0}^{n-1} L^i\right|\cdot|U|\cdot e && \text{因为 } L^n=0\\
&\leqslant \sum_{i=0}^{n-1} |L|^i\cdot|U|\cdot e && \text{由三角不等式}\\
&= (I-|L|)^{-1}\cdot|U|\cdot e && \text{因为 } |L|^n=0,
\end{aligned}
$$

如果能证明更加强的按分量的不等式

$$(I-|L|)^{-1}\cdot|U|\cdot e\leqslant(|L|+|U|)\cdot e \tag{6.24}$$

则不等式(6.23)成立. 因为$(I-|L|)^{-1}=\sum_{i=0}^{n-1}|L|^i$的所有元素是非负的，如果可证明

$$|U|\cdot e\leqslant(I-|L|)\cdot(|L|+|U|)\cdot e=(|L|+|U|-|L|^2-|L|\cdot|U|)\cdot e$$

或

287

$$0\leqslant(|L|-|L|^2-|L|\cdot|U|)\cdot e=|L|\cdot(I-|L|-|U|)\cdot e, \tag{6.25}$$

则不等式(6.24)成立. 因为$|L|$的所有元素非负，如果能证明

$$0\leqslant(I-|L|-|U|)\cdot e \text{ 或 } |R_J|\cdot e=(|L|+|U|)e\leqslant e, \tag{6.26}$$

则不等式(6.25)成立. 最后，由设$\||R_J|\cdot e\|_\infty=\|R_J\|_\infty=\rho<1$，故不等式(6.26)成立. □

当A严格列对角占优(即A^T严格行对角占优)时，有类似的结论.

读者可容易查证这个简单的准则未应用于模型问题，故我们有必要减弱严格对角占优的假设. 为此需要重视矩阵的图性质(graph property).

定义6.7　*如果不存在置换阵P使得*

$$PAP^T=\left[\begin{array}{c|c}A_{11}&A_{12}\\\hline 0&A_{22}\end{array}\right].$$

则A为一个不可约矩阵.

下面把这个定义与图论(graph theory)联系起来.

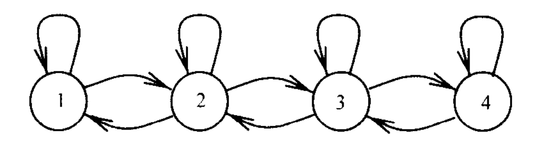
定义6.8　*方向图是由一组有限条有向边(即从一个节点到另一个节点的箭)连接的一组有限个节点. 在方向图中的路径是节点$n_1,\cdots,n_m$的序列，从每个n_i到n_{i+1}有一条边. 自边 (self edge) 是一个节点到它自身的一条边.*

定义6.9　*A的方向图$G(A)$是具有节点$1,2\cdots,n$的图并且从节点i到节点j有一条边当且仅当$a_{ij}\neq0$.*

例6.1　矩阵

$$A=\begin{bmatrix}2&-1&&\\-1&2&-1&\\&-1&2&-1\\&&-1&2\end{bmatrix}$$

有方向图



◇

定义 6.10 如果从每个节点 i 到每个节点 j 都存在一条路径，则方向图称为强连接的(strongly connected). 一个方向图的强连接部分是一个子图(节点的一个子集，这些节点都用边相连接)这个子图是强连接的并且不能作出更大的但仍然是强连接的子图.

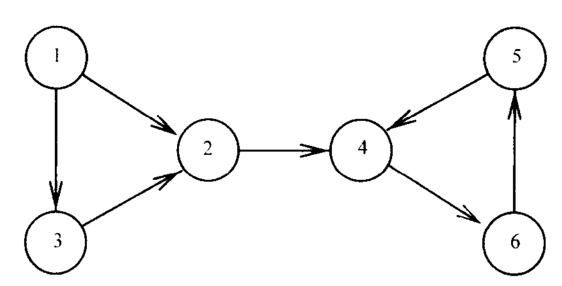
288

例 6.2 例 6.1 中的图是强连接的. ◇

例 6.3 设

$$A = \left[\begin{array}{ccc|ccc} & 1 & 1 & & & \\ & & & 1 & & \\ & 1 & & & & \\ \hline & & & & & 1 \\ & & & 1 & & \\ & & & & 1 & \end{array}\right],$$

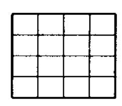
它有方向图



此图不是强连接的，因为从任何节点到节点 1 都不存在路径. 节点 4、5 和 6 构成一个强连接部分，因为从它们中的任一个节点到任何其他节点都存在一条路径.

◇

例 6.4 模型问题的图是强连接的. 图基本上是下列形式：



除了网格中的每条边表示两条边(每个方向一条)外，并且自边未指出. ◇

引理 6.6 A 是不可约的当且仅当 $G(A)$ 是强连接的.

证明 若 $A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$ 可约，则显然无法使得从对应于 A_{22} 的节点后退至对应于 A_{11}

的一个节点，即 $G(\boldsymbol{A})$ 不是强连接的. 类似地，若 $G(\boldsymbol{A})$ 不是强连接的，对行(和列)重新编号以致一个特别的强连接部分中的所有节点首先出现，于是矩阵 $\boldsymbol{PAP}^{\mathrm{T}}$ 将是块上三角阵. □

例6.5 例6.3中的矩阵 $\boldsymbol{A}$ 是可约的.

定义6.11 若对一切 i，$|a_{ii}| \geqslant \sum_{k\neq i}|a_{ik}|$ 并且至少一个是严格不等式，则 $\boldsymbol{A}$ 是弱行

289

对角占优的.

定理6.3 若 $\boldsymbol{A}$ 不可约且弱行对角占优，则雅可比法和高斯-塞德尔法都收敛，且 $\rho(\boldsymbol{R}_{GS}) < \rho(\boldsymbol{R}_J) < 1$.

本定理的证明见[249].

例6.6 模型问题是弱对角占优且不可约的，但不是强对角占优的(对角元是4，而非对角元之和是2或者是3或者是4). 故对模型问题雅可比法和高斯-塞德尔法收敛. ◇

尽管上面的结果表明在一定的条件下高斯-塞德尔法比雅可比法收敛快，但是这类结论一般不成立. 这是因为存在一些非对称阵，对这些矩阵雅可比法收敛而高斯-塞德尔法发散，同样存在一些矩阵，对它们高斯-塞德尔法收敛而雅可比法发散[249].

现在考虑 SOR(ω) 的收敛性[249]. 回顾它的定义:

$$\boldsymbol{R}_{\mathrm{SOR}(\omega)} = (\boldsymbol{I} - \omega \boldsymbol{L})^{-1}((1-\omega)\boldsymbol{I} + \omega \boldsymbol{U}).$$

定理6.4 $\rho(\boldsymbol{R}_{\mathrm{SOR}(\omega)}) \geqslant |\omega - 1|$. 所以为了收敛需要 $0 < \omega < 2$.

证明 记 $\boldsymbol{R}_{\mathrm{SOR}(\omega)}$ 的特征多项式为 $\varphi(\lambda) = \det(\lambda \boldsymbol{I} - \boldsymbol{R}_{\mathrm{SOR}(\omega)}) = \det((\boldsymbol{I} - \omega\boldsymbol{L})(\lambda\boldsymbol{I} - \boldsymbol{R}_{\mathrm{SOR}(\omega)})) = \det((\lambda + \omega - 1)\boldsymbol{I} - \omega\lambda\boldsymbol{L} - \omega\boldsymbol{U})$，所以

$$\varphi(0) = \pm \prod_{i=1}^{n} \lambda_i(\boldsymbol{R}_{\mathrm{SOR}(\omega)}) = \pm \det((\omega - 1)\boldsymbol{I}) = \pm(\omega - 1)^n,$$

推得 $\max_i |\lambda_i(\boldsymbol{R}_{\mathrm{SOR}(\omega)})| \geqslant |\omega - 1|$. □

定理6.5 若 $\boldsymbol{A}$ 对称正定，则对一切 $0 < \omega < 2$，$\rho(\boldsymbol{R}_{\mathrm{SOR}(\omega)}) < 1$，故对一切 $0 < \omega < 2$，SOR(ω) 收敛. 取 $\omega = 1$，可看出高斯-塞德尔法也收敛.

证明 有两个步骤. 简化 $\boldsymbol{R}_{\mathrm{SOR}(\omega)} = \boldsymbol{R}$. 利用(6.19)式的记号，设 $\boldsymbol{M} = \omega^{-1}(\boldsymbol{D} - \omega \tilde{\boldsymbol{L}})$. 则

(1) 定义 $\boldsymbol{Q} = \boldsymbol{A}^{-1}(2\boldsymbol{M} - \boldsymbol{A})$ 并对一切 i 证明 $\Re\lambda_i(\boldsymbol{Q}) > 0$.

(2) 证明 $\boldsymbol{R} = (\boldsymbol{Q} - \boldsymbol{I})(\boldsymbol{Q} + \boldsymbol{I})^{-1}$，推得对一切 i，$|\lambda_i(R)| < 1$.

对(1)，注意 $\boldsymbol{Q}\boldsymbol{x} = \lambda\boldsymbol{x}$ 推得 $(2\boldsymbol{M} - \boldsymbol{A})\boldsymbol{x} = \lambda\boldsymbol{A}\boldsymbol{x}$ 或 $\boldsymbol{x}^*(2\boldsymbol{M} - \boldsymbol{A})\boldsymbol{x} = \lambda\boldsymbol{x}^*\boldsymbol{A}\boldsymbol{x}$. 把上式和它的共轭转置相加得到 $\boldsymbol{x}^*(\boldsymbol{M} + \boldsymbol{M}^* - \boldsymbol{A})\boldsymbol{x} = (\Re\lambda)(\boldsymbol{x}^*\boldsymbol{A}\boldsymbol{x})$. 故由 $\boldsymbol{A}$ 和 $(\frac{2}{\omega} - 1)\boldsymbol{D}$ 正

290

定得 $\Re\lambda = \boldsymbol{x}^*(\boldsymbol{M} + \boldsymbol{M}^* - \boldsymbol{A})\boldsymbol{x}/\boldsymbol{x}^*\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^*(\frac{2}{\omega} - 1)\boldsymbol{D}\boldsymbol{x}/\boldsymbol{x}^*\boldsymbol{A}\boldsymbol{x} > 0$.

为证明(2)，注意 $(\boldsymbol{Q} - \boldsymbol{I})(\boldsymbol{Q} + \boldsymbol{I})^{-1} = (2\boldsymbol{A}^{-1}\boldsymbol{M} - 2\boldsymbol{I})(2\boldsymbol{A}^{-1}\boldsymbol{M})^{-1} = \boldsymbol{I} - \boldsymbol{M}^{-1}\boldsymbol{A} = \boldsymbol{R}$，故由谱映射定理(问题4.5)知

$$|\lambda(\boldsymbol{R})| = \left|\frac{\lambda(\boldsymbol{Q})-1}{\lambda(\boldsymbol{Q})+1}\right| = \left|\frac{(\Re\lambda(\boldsymbol{Q})-1)^2+(\Im\lambda(\boldsymbol{Q}))^2}{(\Re\lambda(\boldsymbol{Q})+1)^2+(\Im\lambda(\boldsymbol{Q}))^2}\right|^{1/2} < 1. \qquad \square$$

定理6.4和6.5合在一起推出，若$\boldsymbol{A}$对称正定，则SOR(ω)收敛当且仅当$0<\omega<2$.

例6.7 模型问题是对称正定的，所以对$0<\omega<2$，SOR(ω)收敛. ◇

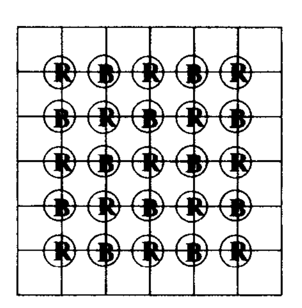
为了最终对模型问题比较雅可比法，高斯-塞德尔法和SOR(ω)法的代价，我们对$\boldsymbol{A}$强加另一个图论的条件. 这个条件通常产生于某种离散的诸如泊松方程那样的偏微分方程. 这个条件将让我们依据$\rho(\boldsymbol{R}_J)$明确地计算$\rho(\boldsymbol{R}_{GS})$和$\rho(\boldsymbol{R}_{\mathrm{SOR}(\omega)})$.

定义6.12 如果存在一个置换阵$\boldsymbol{P}$使得

$$\boldsymbol{PTP}^{\mathrm{T}} = \left[\begin{array}{c|c} \boldsymbol{T}_{11} & \boldsymbol{T}_{12} \\ \hline \boldsymbol{T}_{21} & \boldsymbol{T}_{22} \end{array}\right],$$

其中$\boldsymbol{T}_{11}$和$\boldsymbol{T}_{22}$是对角阵，则称$\boldsymbol{T}$有性质A. 换言之，在图$G(A)$中节点分成两个集合$S_1 \cup S_2$，其中同时在S_1中或同时在S_2中的两个节点之间不存在边（不计自边）. 这种图称为分为两部分的.

例6.8 模型问题的红-黑次序. 这是在6.5.2节中引入的，利用下列类似像棋盘的图形描述模型问题的图. Ⓑ节点在S_1中，而Ⓡ节点在S_2中（Ⓑ表示黑节点，Ⓡ表示红节点）.



正如6.5.2节中所述，模型问题中的每个方程把一个网格点上的值与它的左、右、上和下相邻的点上的值联系起来，这四个点的颜色不同于在中间的网格点颜色. 换言之，从一个Ⓡ结点到一个Ⓡ结点或从一个Ⓑ结点到一个Ⓑ结点不存在直接的连接. 故若我们在黑色的结点之前对红色的结点编号，则矩阵将具有定义6.2要求的形式. 例如，在3×3网格的情况，我们得到下列： 291

$$\boldsymbol{P}\left[\begin{array}{ccc|ccc|ccc}
4 & -1 & & -1 & & & & & \\
-1 & 4 & -1 & & -1 & & & & \\
& -1 & 4 & & & -1 & & & \\ \hline
-1 & & & 4 & -1 & & -1 & & \\
& -1 & & -1 & 4 & -1 & & -1 & \\
& & -1 & & -1 & 4 & & & -1 \\ \hline
& & & -1 & & & 4 & -1 & \\
& & & & -1 & & -1 & 4 & -1 \\
& & & & & -1 & & -1 & 4
\end{array}\right]\boldsymbol{P}^{\mathrm{T}}$$

$$
=\left[\begin{array}{ccccc|cccc}
4 & & & & & -1 & -1 & & \\
 & 4 & & & & -1 & & -1 & \\
 & & 4 & & & -1 & -1 & -1 & -1 \\
 & & & 4 & & & -1 & & -1 \\
 & & & & 4 & & & -1 & -1 \\
\hline
-1 & -1 & -1 & & & 4 & & & \\
-1 & & -1 & -1 & & & 4 & & \\
 & -1 & -1 & & -1 & & & 4 & \\
 & & -1 & -1 & -1 & & & & 4
\end{array}\right]. \qquad \diamond
$$

现在假设T有性质A，因而可记(其中$\boldsymbol{D}_i=\boldsymbol{T}_{ii}$为对角阵)

$$
\boldsymbol{PTP}^{\mathrm{T}}=\begin{bmatrix}\boldsymbol{D}_1 & \boldsymbol{T}_{12}\\ \boldsymbol{T}_{21} & \boldsymbol{D}_2\end{bmatrix}=\begin{bmatrix}\boldsymbol{D}_1 & \\ & \boldsymbol{D}_2\end{bmatrix}-\begin{bmatrix}0 & 0\\ -\boldsymbol{T}_{21} & 0\end{bmatrix}-\begin{bmatrix}0 & -\boldsymbol{T}_{12}\\ 0 & 0\end{bmatrix}=\boldsymbol{D}-\widetilde{\boldsymbol{L}}-\widetilde{\boldsymbol{U}}.
$$

定义6.13　设$R_J(\alpha)=\alpha\boldsymbol{L}+\frac{1}{\alpha}\boldsymbol{U}$. 则$\boldsymbol{R}_J(1)=\boldsymbol{R}_J$是雅可比法的迭代矩阵.

命题6.2　$\boldsymbol{R}_J(\alpha)$的特征值与α无关.

证明

$$
\boldsymbol{R}_J(\alpha)=-\begin{bmatrix}0 & \frac{1}{\alpha}\boldsymbol{D}_1^{-1}\boldsymbol{T}_{12}\\ \alpha\boldsymbol{D}_2^{-1}\boldsymbol{T}_{21} & 0\end{bmatrix}
$$

与相似矩阵

$$
\begin{bmatrix}\boldsymbol{I} & \\ & \alpha\boldsymbol{I}\end{bmatrix}^{-1}\boldsymbol{R}_J(\alpha)\begin{bmatrix}\boldsymbol{I} & \\ & \alpha\boldsymbol{I}\end{bmatrix}=-\begin{bmatrix}0 & \boldsymbol{D}_1^{-1}\boldsymbol{T}_{12}\\ \boldsymbol{D}_2^{-1}\boldsymbol{T}_{21} & 0\end{bmatrix}=\boldsymbol{R}_J(1).
$$

有相同的特征值. □

定义6.14　设$\boldsymbol{T}$是任意的矩阵，$\boldsymbol{T}=\boldsymbol{D}-\widetilde{\boldsymbol{L}}-\widetilde{\boldsymbol{U}}$以及$\boldsymbol{R}_J(\alpha)=\alpha\boldsymbol{D}^{-1}\widetilde{\boldsymbol{L}}+\frac{1}{\alpha}\boldsymbol{D}^{-1}\widetilde{\boldsymbol{U}}$.

292　若$\boldsymbol{R}_J(\alpha)$的特征值与α无关，则称T有相容次序(consistent ordering).

下面是一个简单的事实；若$\boldsymbol{T}$有性质A(例如模型问题)，则对使$\boldsymbol{PTP}^{\mathrm{T}}=\begin{bmatrix}\boldsymbol{T}_{11} & \boldsymbol{T}_{12}\\ \boldsymbol{T}_{21} & \boldsymbol{T}_{22}\end{bmatrix}$有对角阵$\boldsymbol{T}_{11}$和$\boldsymbol{T}_{22}$的置换阵$\boldsymbol{P}$，$\boldsymbol{PTP}^{\mathrm{T}}$有相容次序. 而有相容次序推出矩阵有性质$A$是不成立的.

例6.9　当$\boldsymbol{D}_i$是对角阵时，任何块三对角阵

$$
\begin{bmatrix}\boldsymbol{D}_1 & \boldsymbol{A}_1 & & \\ \boldsymbol{B}_1 & \ddots & \ddots & \\ & \ddots & \ddots & \boldsymbol{A}_{n-1}\\ & & \boldsymbol{B}_{n-1} & \boldsymbol{D}_n\end{bmatrix}
$$

有相容次序. ◇

相容次序推出存在一些与 $\boldsymbol{R}_J$, $\boldsymbol{R}_{GS}$ 和 $\boldsymbol{R}_{\mathrm{SOR}(\omega)}$ 之特征值相关的简单公式[249].

定理6.6 若 $\boldsymbol{A}$ 有相容次序且 $\omega\neq0$，则下列命题成立：

1）$\boldsymbol{R}_J$ 的特征值按 ± 成对出现.

2）若 μ 是 $\boldsymbol{R}_J$ 的一个特征值且

$$(\lambda+\omega-1)^2=\lambda\omega^2\mu^2, \tag{6.27}$$

则 λ 是 $R_{\mathrm{SOR}(\omega)}$ 的一个特征值.

3）反之，若 $\lambda\neq0$ 是 $\boldsymbol{R}_{\mathrm{SOR}(\omega)}$ 的一个特征值，则(6.27)式中的 μ 是 $\boldsymbol{R}_J$ 的一个特征值.

证明

(1)由相容次序推出 $R_J(\alpha)$ 的特征值与 α 无关，故 $\boldsymbol{R}_J=\boldsymbol{R}_J(1)$ 和 $\boldsymbol{R}_J(-1)=-\boldsymbol{R}_J(1)$ 有相同的特征值. 因此它们按 ± 成对出现.

(2)若 $\lambda=0$ 且(6.27)式成立，则 $\omega=1$ 且 0 确实是 $\boldsymbol{R}_{\mathrm{SOR}(1)}=\boldsymbol{R}_{GS}=(\boldsymbol{I}-\boldsymbol{L})^{-1}\boldsymbol{U}$ 的一个特征值，因为 $\boldsymbol{R}_{GS}$ 奇异. 否则

$$\begin{aligned}
0&=\det(\lambda\boldsymbol{I}-\boldsymbol{R}_{\mathrm{SOR}(\omega)})\\
&=\det((\boldsymbol{I}-\omega\boldsymbol{L})(\lambda\boldsymbol{I}-\boldsymbol{R}_{\mathrm{SOR}(\omega)}))\\
&=\det((\lambda+\omega-1)\boldsymbol{I}-\omega\lambda\boldsymbol{L}-\omega\boldsymbol{U})\\
&=\det\left(\sqrt{\lambda}\omega\left(\left(\frac{\lambda+\omega-1}{\sqrt{\lambda}\omega}\right)\boldsymbol{I}-\sqrt{\lambda}\boldsymbol{L}-\frac{1}{\sqrt{\lambda}}\boldsymbol{U}\right)\right)\\
&=\det\left(\left(\frac{\lambda+\omega-1}{\sqrt{\lambda}\omega}\right)\boldsymbol{I}-\boldsymbol{L}-\boldsymbol{U}\right)(\sqrt{\lambda}\omega)^n,
\end{aligned}$$

其中最后的等式成立是因为命题6.2. 所以 $\dfrac{\lambda+\omega-1}{\sqrt{\lambda}\omega}=\mu$ 是 $\boldsymbol{L}+\boldsymbol{U}=\boldsymbol{R}_J$ 的一个特征值，且 $(\lambda+\omega-1)^2=\mu^2\omega^2\lambda$.

293

(3)若 $\lambda\neq0$，则最后的一组等式按反向操作. □

推论6.1 若 $\boldsymbol{A}$ 有相容次序，则 $\rho(\boldsymbol{R}_{GS})=(\rho(\boldsymbol{R}_J))^2$. 这意味着高斯-塞德尔法的速度是雅可比法的二倍.

证明 选择 $\omega=1$ 等价于高斯-塞德尔法，因而 $\lambda^2=\lambda\mu^2$ 或 $\lambda=\mu^2$. □

为了从超松弛得到最大的好处，我们将求出使 $\rho(\boldsymbol{R}_{\mathrm{SOR}(\omega)})$ 达到极小的 ω_{opt}. [249].

定理6.7 假定 $\boldsymbol{A}$ 有相容次序，$\boldsymbol{R}_J$ 有实特征值，且 $\mu=\rho(\boldsymbol{R}_J)<1$. 则

$$\omega_{\mathrm{opt}}=\frac{2}{1+\sqrt{1-\mu^2}},$$

$$\rho(\boldsymbol{R}_{\mathrm{SOR}(\omega_{\mathrm{opt}})})=\omega_{\mathrm{opt}}-1=\frac{\mu^2}{[1+\sqrt{1-\mu^2}]^2},$$

$$\rho(\boldsymbol{R}_{\mathrm{SOR}(\omega)})=\begin{cases}\omega-1, & \omega_{\mathrm{opt}}\leqslant\omega\leqslant2,\\ 1-\omega+\dfrac{1}{2}\omega^2\mu^2+\omega\mu\sqrt{1-\omega+\dfrac{1}{4}\omega^2\mu^2}, & 0<\omega\leqslant\omega_{\mathrm{opt}}\end{cases}$$

证明　解$(\lambda+\omega-1)^2=\lambda\omega^2\mu^2$得到$\lambda$. □

例6.10　模型问题是一个例子：$\boldsymbol{R}_J$对称，故它有实特征值. 图6-5对$N=16$和$N=64$的$N\times N$网格上的模型问题显示相对于ω的$\rho(\boldsymbol{R}_{\mathrm{SOR}(\omega)})$以及$\rho(\boldsymbol{R}_{GS})$和$\rho(\boldsymbol{R}_J)$的图形. 左边的图形是$\rho(\boldsymbol{R})$的图形而右边的图形是$1-\rho(\boldsymbol{R})$的半对数图形. 可以获得的主要结论是$\rho(\boldsymbol{R}_{\mathrm{SOR}(\omega)})$的图像有一个非常狭窄的极小，因而若$\omega$即使稍微不同于$\omega_{\mathrm{opt}}$，则收敛将明显地慢下来. 第二个结论是若你不得不猜测ω_{opt}，则猜测大的值(接近于2)比猜测小的值更好. ◇

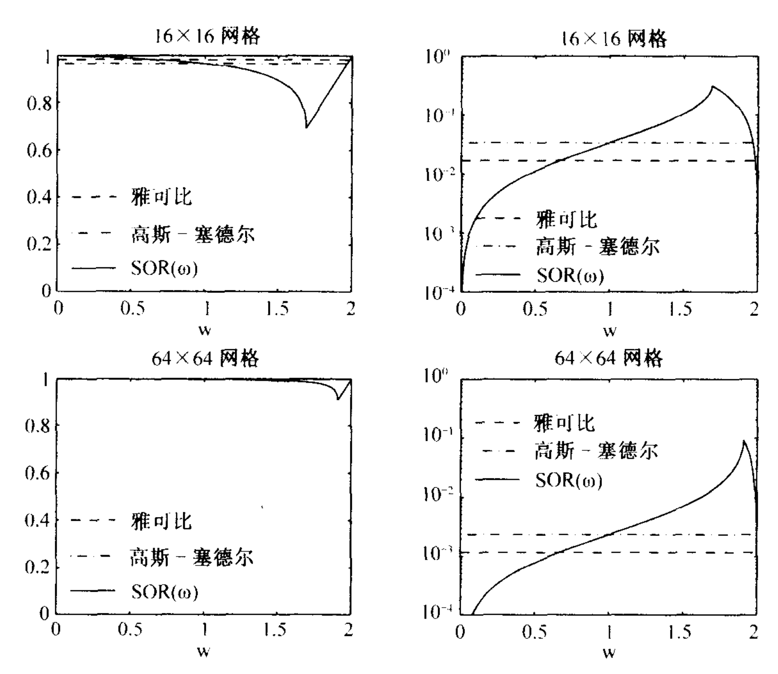


图6-5　在16×16网格和64×64网格上对模型问题雅可比法，高斯-塞德尔法以及相对于ω的SOR(ω)法的收敛性. 每个方法的谱半径$\rho(R_J)$，$\rho(R_{\mathrm{SOR}(\omega)})$，$\rho(R_{GS})$和$\rho(R)$画在左边，而$1-\rho(R)$画在右边

6.5.6　切比雪夫加速和对称SOR(SSOR)

迄今为止我们已讨论的方法中，雅可比法和高斯-塞德尔法不需要有关执行它们的矩阵的信息(虽然证明它们收敛需要某些信息). SOR(ω)依赖于参数ω，根据$\rho(\boldsymbol{R}_J)$来选择这个参数ω去加速收敛. 当知道比$\rho(\boldsymbol{R}_J)$更多的有关$\boldsymbol{R}_J$的谱时切比雪夫加速是有用的，并且可进一步加速收敛.

假如利用某种方法(雅可比法、高斯-塞德尔法或SOR(ω)法)把$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$转换成迭代$\boldsymbol{x}_{i+1}=\boldsymbol{R}\boldsymbol{x}_i+c$，则当$\rho(\boldsymbol{R})<1$时得到一个序列$\{x_i\}$，当$i\to\infty$时，$x_i\to x$.

给定所有的这些近似值x_i，自然地要问它们的某个线性组合$y_m=\sum_{i=0}^{m}\gamma_{mi}x_i$是

否为解 $\boldsymbol{x}$ 的一个更好的近似值. 注意标量 γ_{mi} 必须满足 $\sum_{i=0}^{m}\gamma_{mi}=1$，因为若 $\boldsymbol{x}_0=\boldsymbol{x}_1=\cdots=\boldsymbol{x}$，我们也要求 $\boldsymbol{y}_m=\boldsymbol{x}$. 因而可记 $\boldsymbol{y}_m$ 中的误差为

$$\begin{aligned}\boldsymbol{y}_m-\boldsymbol{x} &= \sum_{i=0}^{m}\gamma_{mi}\boldsymbol{x}_i-\boldsymbol{x}\\ &= \sum_{i=0}^{m}\gamma_{mi}(\boldsymbol{x}_i-\boldsymbol{x})\\ &= \sum_{i=0}^{m}\gamma_{mi}\boldsymbol{R}^i(\boldsymbol{x}_0-\boldsymbol{x})\\ &= p_m(\boldsymbol{R})(\boldsymbol{x}_0-\boldsymbol{x}),\end{aligned} \tag{6.28}$$

其中 $p_m(\boldsymbol{R})=\sum_{i=0}^{m}\gamma_{mi}\boldsymbol{R}^i$ 是 m 次多项式且 $p_m(1)=\sum_{i=0}^{m}\gamma_{mi}=1$.

294 ~ 295

例 6.11 若可选择 p_m 是 $\boldsymbol{R}$ 的特征多项式，则由凯莱 - 哈密顿定理知 $p_m(\boldsymbol{R})=0$，而且 m 步将收敛. 但这是不实用的，因为我们很少知道 $\boldsymbol{R}$ 的特征值，并且无论如何希望收敛快于 $m=\dim(\boldsymbol{R})$ 步. ◇

代替寻找使 $p_m(\boldsymbol{R})$ 为零的多项式，我们将满足于使 $p_m(\boldsymbol{R})$ 的谱半径如我们尽可能达到的那样小. 假如知道.

- $\boldsymbol{R}$ 的特征值是实的，且
- $\boldsymbol{R}$ 的特征值位于一个不包含 1 的区间 $[-\rho,\rho]$ 中.

则我们可尝试选择一个多项式 p_m，其中

1) $p_m(1)=1$，且

2) $\max_{-\rho\leqslant x\leqslant\rho}|p_m(x)|$ 尽可能小.

因为 $p_m(\boldsymbol{R})$ 的特征值是 $p_m(\lambda(\boldsymbol{R}))$（见问题 4.5），所以这些特征值将是小的，因而谱半径（绝对值最大的特征值）将是小的.

求满足上面的条件 1）和 2）的多项式 p_m 是逼近论中的一个经典问题，它的解以切比雪夫多项式为基础.

定义 6.15 m 次切比雪夫多项式是由递推 $T_m(x)\equiv 2xT_{m-1}(x)-T_{m-2}(x)$ 定义的，其中 $T_0(x)=1$ 且 $T_1(x)=x$.

切比雪夫多项式有许多有趣的性质[240]. 下面是利用定义容易证明的一些性质（见问题 6.7）.

引理 6.7 切比雪夫多项式有下列性质:

- $T_m(1)=1$.
- $T_m(x)=2^{m-1}x^m+O(x^{m-1})$.
- $T_m(x)=\begin{cases}\cos(m\cdot\arccos x) & 若\ |x|\leqslant 1,\\ \cosh(m\cdot\operatorname{arccosh} x) & 若\ |x|\geqslant 1.\end{cases}$
- $|T_m(x)|\leqslant 1$ 　若 $|x|\leqslant 1$.
- $T_m(x)$ 的零点是 $x_i=\cos((2i-1)\pi/(2m)),i=1,\cdots,m$.

- $T_m(x)=\frac{1}{2}[(x+\sqrt{x^2-1})^m+(x+\sqrt{x^2-1})^{-m}]$　　若 $|x|>1$.
- $T_m(1+\varepsilon)\geq .5(1+m\sqrt{2\varepsilon})$　　若 $\varepsilon>0$.

下面是 $T_m(1+\varepsilon)$ 的一个值表. 注意即使当 ε 是微小的时候，它如何随 m 增长而 296 快速增长(见图6-6).

m	ε		
	10^{-4}	10^{-3}	10^{-2}
10	1.0	1.1	2.2
100	2.2	44	$6.9\cdot10^5$
200	8.5	$3.8\cdot10^3$	$9.4\cdot10^{11}$
1000	$6.9\cdot10^5$	$1.3\cdot10^{19}$	$1.2\cdot10^{61}$

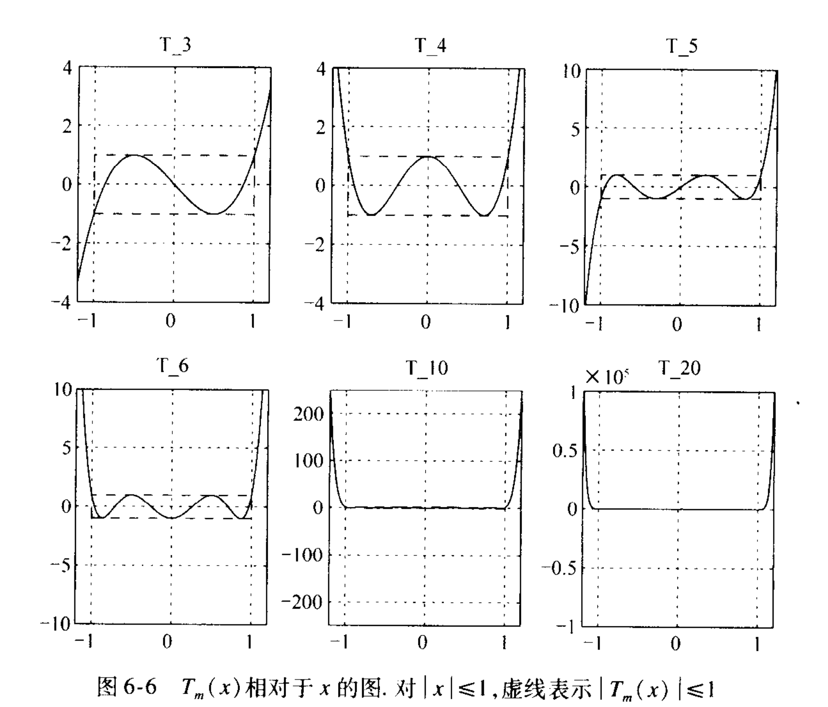


图6-6　$T_m(x)$ 相对于 x 的图. 对 $|x|\leq1$, 虚线表示 $|T_m(x)|\leq1$

我们想要的具有这些性质的多项式是 $p_m(x)=T_m(x/\rho)/T_m(1/\rho)$. 为观察原因，注意 $p_m(1)=1$ 并且若 $x\in[-\rho,\rho]$，则 $|p_m(x)|\leq1/T_m(1/\rho)$. 例如，若 $\rho=1/(1+\varepsilon)$，则 $|p_m(x)|\leq1/T_m(1+\varepsilon)$. 正如刚才已看到的，对小的 ε 和中等大小的 m，这个界是微小的.

为廉价地执行这个计算，利用定义切比雪夫多项式的三项递推 $T_m(x)=2xT_{m-1}(x)-T_{m-2}(x)$. 这意味着只有三个向量 y_m, y_{m-1} 和 y_{m-2} 不是所有前面的 x_m 需要存储和组

合. 为观察如何操作，设 $\mu_m \equiv 1/T_m(1/\rho)$，因而 $p_m(\boldsymbol{R}) = \mu_m \boldsymbol{T}_m(\boldsymbol{R}/\rho)$ 且由定义 6.15 中的三项递推知 $\frac{1}{\mu_m} = \frac{2}{\rho\mu_{m-1}} - \frac{1}{\mu_{m-2}}$. 于是

$$\begin{aligned}
\boldsymbol{y}_m - \boldsymbol{x} &= p_m(\boldsymbol{R})(\boldsymbol{x}_0 - \boldsymbol{x}) \quad \text{由(6.28)式} \\
&= \mu_m \boldsymbol{T}_m\left(\frac{\boldsymbol{R}}{\rho}\right)(\boldsymbol{x}_0 - \boldsymbol{x}) \\
&= \mu_m\left[2 \cdot \frac{\boldsymbol{R}}{\rho} \cdot \boldsymbol{T}_{m-1}\left(\frac{\boldsymbol{R}}{\rho}\right)(\boldsymbol{x}_0 - \boldsymbol{x}) - \boldsymbol{T}_{m-2}\left(\frac{\boldsymbol{R}}{\rho}\right)(\boldsymbol{x}_0 - \boldsymbol{x})\right] \quad \text{由定义 6.15} \\
&= \mu_m\left[2 \cdot \frac{\boldsymbol{R}}{\rho} \cdot \frac{p_{m-1}\left(\frac{\boldsymbol{R}}{\rho}\right)(\boldsymbol{x}_0 - \boldsymbol{x})}{\mu_{m-1}} - \frac{p_{m-2}\left(\frac{\boldsymbol{R}}{\rho}\right)(\boldsymbol{x}_0 - \boldsymbol{x})}{\mu_{m-2}}\right] \\
&= \mu_m\left[2 \cdot \frac{\boldsymbol{R}}{\rho} \cdot \frac{\boldsymbol{y}_{m-1} - \boldsymbol{x}}{\mu_{m-1}} - \frac{\boldsymbol{y}_{m-2} - \boldsymbol{x}}{\mu_{m-2}}\right] \quad \text{由(6.28)式}
\end{aligned}$$

或

$$\boldsymbol{y}_m = \frac{2\mu_m}{\mu_{m-1}} \frac{\boldsymbol{R}}{\rho} \boldsymbol{y}_{m-1} - \frac{\mu_m}{\mu_{m-2}} \boldsymbol{y}_{m-2} + \boldsymbol{d}_m,$$

其中

$$\begin{aligned}
\boldsymbol{d}_m &= \boldsymbol{x} - \frac{2\mu_m}{\mu_{m-1}}\left(\frac{\boldsymbol{R}}{\rho}\right)\boldsymbol{x} + \frac{\mu_m}{\mu_{m-2}}\boldsymbol{x} \\
&= \boldsymbol{x} - \frac{2\mu_m}{\mu_{m-1}}\left(\frac{\boldsymbol{x} - \boldsymbol{c}}{\rho}\right) + \frac{\mu_m}{\mu_{m-2}}\boldsymbol{x} \quad \text{因为 } \boldsymbol{x} = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{c} \\
&= \mu_m\left(\frac{1}{\mu_m} - \frac{2}{\rho\mu_{m-1}} + \frac{1}{\mu_{m-2}}\right)\boldsymbol{x} + \frac{2\mu_m}{\rho\mu_{m-1}}\boldsymbol{c} \\
&= \frac{2\mu_m}{\rho\mu_{m-1}}\boldsymbol{c} \quad \text{由 } \mu_m \text{ 的定义}
\end{aligned}$$

这就得到算法.

算法 6.7 $\boldsymbol{x}_{i+1} = \boldsymbol{R}\boldsymbol{x}_i + \boldsymbol{c}$ 的切比雪夫加速：

$\mu_0 = 1$; $\mu_1 = \rho$; $\boldsymbol{y}_0 = \boldsymbol{x}_0$; $\boldsymbol{y}_1 = \boldsymbol{R}\boldsymbol{x}_0 + \boldsymbol{c}$

for $m = 2, 3, \cdots$

$$\mu_m = 1/\left(\frac{2}{\rho\mu_{m-1}} - \frac{1}{\mu_{m-2}}\right)$$

$$y_m = \frac{2\mu_m}{\rho\mu_{m-1}}\boldsymbol{R}\boldsymbol{y}_{m-1} - \frac{\mu_m}{\mu_{m-2}}\boldsymbol{y}_{m-2} + \frac{2\mu_m}{\rho\mu_{m-1}}\boldsymbol{c}$$

end for

注意每个迭代应用 $\boldsymbol{R}$ 仅仅一次，因而如果这个运算与其他的标量和向量运算相比相当昂贵，则这个算法每步与原来的迭代 $\boldsymbol{x}_{m+1} = \boldsymbol{R}\boldsymbol{x}_m + \boldsymbol{c}$ 相比不昂贵.

297

遗憾地，我们不能直接地对求解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的 SOR(ω)应用这个加速，因为 $\boldsymbol{R}_{\mathrm{SOR}(\omega)}$ 一般有复特征值，而切比雪夫加速需要区间$[-\rho,\ \rho]$中 R 有实特征值. 但是可利用下列算法确定这个值.

298

算法 6.8 对称 SOR(SSOR)：

1. 作 SOR(ω)的单步按通常的递增次序计算 x 的分量：$x_{i,1},x_{i,2},\cdots,x_{i,n}$；
2. 作 SOR(ω)的单步反向计算：$x_{i,n},x_{i,n-1},\cdots,x_{i,1}$.

我们将重新表达这个算法为 $\boldsymbol{x}_{i+1}=\boldsymbol{E}_\omega\boldsymbol{x}_i+\boldsymbol{c}_\omega$，并证明 $\boldsymbol{E}_\omega$ 有实特征值，因而可以使用切比雪夫加速.

假如 $\boldsymbol{A}$ 如模型问题中那样为对称阵，并与(6.19)式中一样再记 $\boldsymbol{A}=\boldsymbol{D}-\widetilde{\boldsymbol{L}}-\widetilde{\boldsymbol{U}}=\boldsymbol{D}(\boldsymbol{I}-\boldsymbol{L}-\boldsymbol{U})$. 因为 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$，所以 $\boldsymbol{U}=\boldsymbol{L}^{\mathrm{T}}$. 利用(6.21)式重写 SSOR 的两步为

1. $\boldsymbol{x}_{i+1/2}=(\boldsymbol{I}-\omega\boldsymbol{L})^{-1}((1-\omega)\boldsymbol{I}+\omega\boldsymbol{U})\boldsymbol{x}_i+c_{1/2}\equiv\boldsymbol{L}_\omega\boldsymbol{x}_i+c_{1/2}$,
2. $\boldsymbol{x}_i=(\boldsymbol{I}-\omega\boldsymbol{U})^{-1}((1-\omega)\boldsymbol{I}+\omega\boldsymbol{L})\boldsymbol{x}_{i+1/2}+c_1\equiv\boldsymbol{U}_\omega\boldsymbol{x}_{i+1/2}+c_1$.

消去 $\boldsymbol{x}_{i+1/2}$ 得到 $\boldsymbol{x}_{i+1}=\boldsymbol{E}_\omega\boldsymbol{x}_i+\hat{c}$，其中

$$\begin{aligned}\boldsymbol{E}_\omega&=\boldsymbol{U}_\omega\boldsymbol{L}_\omega\\&=\boldsymbol{I}+(\omega-2)^2(\boldsymbol{I}-\omega\boldsymbol{U})^{-1}(\boldsymbol{I}-\omega\boldsymbol{L})^{-1}+(\omega-2)(\boldsymbol{I}-\omega\boldsymbol{U})^{-1}\\&\quad+(\omega-2)(\boldsymbol{I}-\omega\boldsymbol{U})^{-1}(\boldsymbol{I}-\omega\boldsymbol{L})^{-1}(\boldsymbol{I}-\omega\boldsymbol{U}).\end{aligned}$$

我们断言 $\boldsymbol{E}_\omega$ 有实特征值，因为它与相似矩阵

$$\begin{aligned}&(\boldsymbol{I}-\omega\boldsymbol{U})\boldsymbol{E}_\omega(\boldsymbol{I}-\omega\boldsymbol{U})^{-1}\\&=\boldsymbol{I}+(2-\omega)^2(\boldsymbol{I}-\omega\boldsymbol{L})^{-1}(\boldsymbol{I}-\omega\boldsymbol{U})^{-1}+(\omega-2)(\boldsymbol{I}-\omega\boldsymbol{U})^{-1}\\&\quad+(\omega-2)(\boldsymbol{I}-\omega\boldsymbol{L})^{-1}\\&=\boldsymbol{I}+(2-\omega)^2(\boldsymbol{I}-\omega\boldsymbol{L})^{-1}(\boldsymbol{I}-\omega\boldsymbol{L}^{\mathrm{T}})^{-1}+(\omega-2)(\boldsymbol{I}-\omega\boldsymbol{L}^{\mathrm{T}})^{-1}\\&\quad+(\omega-2)(\boldsymbol{I}-\omega\boldsymbol{L})^{-1},\end{aligned}$$

有相同的特征值，上面这个矩阵这显然是对称的，故必有实特征值.

例 6.12 对模型问题应用带切比雪夫迭代的 SSOR(ω). 需要同时选择 ω 和估计谱半径 $\rho=\rho(\boldsymbol{E}_\omega)$. 不知道极小化 ρ 的最佳 ω，但 Young [267,137] 曾指出 $\omega=\dfrac{2}{1+[2(1-\rho(R_J))]^{1/2}}$ 是一个好的选择，得到 $\rho(\boldsymbol{E}_\omega)\approx1-\dfrac{\pi}{2N}$. 用切比雪夫加速在第 m 步误差用 $\mu_m\approx\dfrac{1}{T_m\left(1+\dfrac{\pi}{2N}\right)}\leqslant2\Big/\left(1+m\sqrt{\dfrac{\pi}{N}}\right)$ 相乘. 所以，用一个固定的小于 1 的因子减少误差需要 $m=O(N^{1/2})=O(n^{1/4})$ 次迭代. 因为每个迭代具有与 SOR(ω)一次迭代相同的代价 $O(n)$，所以整体的代价是 $O(n^{5/4})$. 这就说明了表 6-1 中带切比雪夫加速的 SSOR 的表列数据.

相反，SOR(ω_{opt})的 m 步之后，误差仅仅减少 $\left(1-\dfrac{\pi}{N}\right)^m$. 例如，考虑 $N=1000$.

则为了去掉一半误差 SOR(ω_{opt})需要 $m=220$ 次迭代，而带切比雪夫加速的SOR(ω_{opt})只需要 $m=17$ 次迭代. ◇

6.6 克雷洛夫子空间方法

这些方法同时用于求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 和求 $\boldsymbol{A}$ 的特征值. 它们假设只借助于给定任何 $\boldsymbol{z}$ 获得 $\boldsymbol{y}=\boldsymbol{Az}$ 的“黑箱”子程序就可获取 $\boldsymbol{A}$(当 $\boldsymbol{A}$ 非对称时或许是 $\boldsymbol{y}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{z}$). 换言之，不使用直接获取或处理矩阵元素. 因为有几个理由说明这是一个合理的假设. 第一，能够对一个稀疏矩阵执行的最便宜的非平凡运算是用一个向量乘它. 若 $\boldsymbol{A}$ 有 m 个非零元，矩阵向量乘法花费 m 次乘法以及(至多)m 次加法，第二，$\boldsymbol{A}$ 可能未被明确地表示成一个矩阵，而仅仅是用来作为计算 $\boldsymbol{Ax}$ 的子程序.

例6.13 假如有一个物理装置，它的性态由一个程序来模拟，这个程序有一个输入参数向量 $\boldsymbol{x}$ 和一个输出参数向量 $\boldsymbol{y}$. 输出 $\boldsymbol{y}$ 可能是一个任意复杂的函数 $y=f(x)$，或许是非线性微分方程的解. 例如，x 可能是描述机翼形状的参数而 $f(x)$ 可能是在机翼上的空气阻力. 通过求解关于机翼上气流的 Navier-Stokes 方程来计算 $f(x)$. 通常的工程设计问题是挑选输入 x 去优化装置状态 $f(x)$，为具体起见，我们假定挑选意味着使 $f(x)$ 尽可能小. 于是问题是努力尽我们所能尝试求解 $f(x)=0$. 为了说明假定 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 是同样维数的向量. 于是牛顿法是一个当然的候选方法，得到迭代 $x^{(m+1)}=x^{(m)}-(\nabla f(x^{(m)}))^{-1}f(x^{(m)})$，其中 $\nabla f(x^{(m)})$ 是 f 在 $x^{(m)}$ 上的雅可比阵. 可以改写这个式子为求解线性方程组 $(\nabla f(x^{(m)}))\cdot\delta^{(m)}=f(x^{(m)})$ 得到 $\delta^{(m)}$，然后计算 $x^{(m+1)}=x^{(m)}-\delta^{(m)}$. 但是，当计算 $f(x^{(m)})$ 已经复杂时如何求解系数阵为 $\nabla f(x^{(m)})$ 线性方程组呢? 结果变成对任意的向量 z 可计算矩阵向量积 $(\nabla f(x))\cdot z$，所以我们可利用克雷洛夫子空间方法求解线性方程组. 计算 $(\nabla f(x))\cdot z$ 的一个方法是用均差(divided difference)或利用泰勒展开看出 $[f(x+hz)-f(x)]/h\approx(\nabla f(x))\cdot z$. 因而，计算 $(\nabla f(x))\cdot z$ 需要二次调用计算 $f(\cdot)$ 的子程序，一次是关于变量 x，一次是关于 $x+hz$. 然而选择 h 得到导数的一个精确的近似值有时是困难的(选择 h 太小，由于舍入导致丧失精度). 计算 $(\nabla f(x))\cdot z$ 的另一种方法实际上是对函数 f 求微分. 若 f 足够简单，则这可以手工完成. 对复杂的 f，编译工具能选取一个(几乎是)任意的计算 $f(x)$ 的子程序，并自动产生计算 $(\nabla f(x))\cdot z$ 的另一个子程序[29]. 这也可通过利用C++ 或 Fortran 90 的运算符重载功能来完成，虽然这样做效率较低.

存在各种不同的克雷洛夫子空间方法. 有的方法适合于非对称阵，而另外一些方法假定对称矩阵或正定矩阵. 某些非对称阵的方法假定 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{z}$ 和 $\boldsymbol{Az}$ 都能计算，这依赖于 $\boldsymbol{A}$ 如何表示，$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{z}$ 可能得到或者不可能得到(见例6.13). 最有效的且最好理解的方法是共轭梯度法(CG)，这个方法仅仅适合于包括模型问题的对称正定阵. 在本章中我们将集中讨论 CG.

已知一个非对称正定的矩阵，从众多有效的方法中挑选一个最佳方法可能是困

299
300

难的. 在6.6.6节中除CG之外，我们将给出其他一些有效方法的简短摘要，以及在哪些情况用哪种方法的意见. 还建议读者更充分地了解 NETLIB/templates 上的在线帮助，这包括一本名录[24]以及使用 Matlab，Fortran 及C++ 的执行过程. 关于克雷洛夫子空间方法当前研究的综述见[15,107,136,214].

在第7章中也将讨论求特征值的克雷洛夫子空间方法.

6.6.1 通过矩阵-向量乘法得到关于 A 的信息

给定向量 $\boldsymbol{b}$ 以及计算 $\boldsymbol{A}\cdot\boldsymbol{x}$ 的子程序，我们能够推导关于 $\boldsymbol{A}$ 的什么信息？最明显的事情是可以计算矩阵-向量积的序列 $\boldsymbol{y}_1=\boldsymbol{b},\boldsymbol{y}_2=\boldsymbol{A}\boldsymbol{y}_1,\boldsymbol{y}_3=\boldsymbol{A}\boldsymbol{y}_2=\boldsymbol{A}^2\boldsymbol{y}_1,\cdots,\boldsymbol{y}_n=\boldsymbol{A}\boldsymbol{y}_{n-1}=\boldsymbol{A}^{n-1}\boldsymbol{y}_1$，其中 $\boldsymbol{A}$ 为 $n\times n$ 阶矩阵. 设 $\boldsymbol{K}=[y_1,y_2,\cdots,y_n]$. 则可记

$$\boldsymbol{A}\cdot\boldsymbol{K}=[\boldsymbol{A}y_1,\cdots,\boldsymbol{A}y_{n-1},\boldsymbol{A}y_n]=[\boldsymbol{y}_2,\cdots,\boldsymbol{y}_n,\boldsymbol{A}^n\boldsymbol{y}_1]. \tag{6.29}$$

注意 $\boldsymbol{A}\cdot\boldsymbol{K}$ 的前 $n-1$ 列是与 $\boldsymbol{K}$ 的尾部 $n-1$ 列左移一个位置相同. 暂时假定 $\boldsymbol{K}$ 非奇异，因而可计算 $c=-\boldsymbol{K}^{-1}\boldsymbol{A}^n\boldsymbol{y}_1$. 于是

$$\boldsymbol{A}\cdot\boldsymbol{K}=\boldsymbol{K}\cdot[\boldsymbol{e}_2,\boldsymbol{e}_3,\cdots,\boldsymbol{e}_n,-\boldsymbol{c}]\equiv\boldsymbol{K}\cdot\boldsymbol{C},$$

其中 $\boldsymbol{e}_i$ 是单位阵的第 i 列，或

$$\boldsymbol{K}^{-1}\boldsymbol{A}\boldsymbol{K}=\boldsymbol{C}=\begin{bmatrix}0&0&\cdots&0&-c_1\\1&0&\cdots&0&-c_2\\0&1&\cdots&\vdots&\vdots\\\vdots&0&\cdots&\vdots&\vdots\\\vdots&\vdots&\cdots&0&\vdots\\\vdots&\vdots&\cdots&1&-c_n\end{bmatrix}.$$

注意 C 是上海森伯格(Hessenberg)阵，事实上，它是一个友阵(见4.5.3节)，这意味着它的特征多项式是 $p(x)=x^n+\sum_{i=1}^{n}c_ix^{i-1}$. 因而，只用矩阵-向量乘法已把 $\boldsymbol{A}$ 化成一个非常简单的形式，并且原则上可通过求 $p(x)$ 的零点来求 $\boldsymbol{A}$ 的特征值.

301 然而，这个简单的形式在实际中并不有效，理由如下：

1. 求 $\boldsymbol{C}$ 需要用 $\boldsymbol{A}$ 作 $n-1$ 次矩阵-向量乘法，然后求解关于 $\boldsymbol{K}$ 的一个线性方程组. 即使 $\boldsymbol{A}$ 稀疏，$\boldsymbol{K}$ 有可能是稠密的，故没有理由期望求解关于 $\boldsymbol{K}$ 的线性方程组比求解原来的问题 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 更容易.

2. K 有可能是非常病态的，故将极其不准确地算出 c. 这是因为算法是执行幂法(算法4.1)去得到 $\boldsymbol{K}$ 的列 $\boldsymbol{y}_i$，所以 $\boldsymbol{y}_i$ 收敛到对应于 $\boldsymbol{A}$ 的最大特征值的一个特征向量. 因而，$\boldsymbol{K}$ 的列趋向于变得越来越平行.

下面将克服这些困难问题：用一个正交阵 $\boldsymbol{Q}$ 代替 $\boldsymbol{K}$ 使得对一切 k，$\boldsymbol{K}$ 和 $\boldsymbol{Q}$ 的前 k 列张成相同的空间. 这个空间称为克雷洛夫子空间. 与 $\boldsymbol{K}$ 相反，$\boldsymbol{Q}$ 是良态的且容易求逆. 而且为了得到($\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 或 $\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}$)的精确解，仅仅需要计算 $\boldsymbol{Q}$ 的前面几列. 实际上，与矩阵维数 n 相比较通常需要非常少的列. 记 $\boldsymbol{K}$ 的 QR 分解 $\boldsymbol{K}=\boldsymbol{Q}\boldsymbol{R}$. 然后

$$K^{-1}AK=(R^{-1}Q^{T})A(QR)=C,$$

推出

$$Q^{T}AQ=RCR^{-1}\equiv H.$$

因为 R 和 R^{-1} 都是上三角而 C 是上海森伯格阵．所以容易证实 $H=RCR^{-1}$ 也是上海森伯格阵(见问题6.11)．换言之，我们通过一个正交变换 Q 把 A 化成上海森伯格形式，(这是4.4.6节中讨论的求非对称阵特征值算法的第一步)．注意，若 A 对称，因而 $Q^{T}AQ=H$ 是对称矩阵，并且它是上海森伯格阵也必须是下海森伯格阵，即它是三对角阵．此时可记 $Q^{T}AQ=T$.

仍需要说明如何一次计算 Q 的一列而不是全部列：设 $Q=[q_1,\cdots,q_n]$．因为 $Q^{T}AQ=H$ 推出 $AQ=QH$，由 $AQ=QH$ 的两边第 j 列相等，得到

$$Aq_j=\sum_{i=1}^{j+1}h_{i,j}q_i.$$

因为 q_i 是规范正交的，所以可以用 q_m^{T} 乘上面等式的两边得到

302

$$q_m^{T}Aq_j=\sum_{i=1}^{j+1}h_{i,j}q_m^{T}q_i=h_{m,j}\qquad 1\leqslant m\leqslant j,$$

因而

$$h_{j+1,j}q_{j+1}=Aq_j-\sum_{i=1}^{j}h_{i,j}q_i.$$

这证明下列算法是恰当的.

算法6.9 (部分)约化成海森伯格形式的 Arnoldi 算法

```
q_1 = b/||b||_2
/* k 是计算的 Q 和 H 的列数 */
for j = 1 to k
    z = A q_j
    for i = 1 to j
        h_{i,j} = q_i^T z
        z = z - h_{i,j} q_i
    end for
    h_{j+1,j} = ||z||_2
    if h_{j+1,j} = 0, quit
    q_{j+1} = z/h_{j+1,j}
end for
```

用阿诺尔迪算法计算的 q_j 通常称为阿诺尔迪向量．关于 i 的循环更新 z 也可以描述为应用修正的格拉姆－施密特算法(算法3.1)在从 q_1 到 q_j 方向的分量中减去 z,

保持 z 正交于它们. 计算 q_1 到 q_k 花费 k 次用 A 的矩阵-向量乘法，加上 $O(k^2n)$ 次其他操作. 若在此停止算法，试问获悉了关于 A 的什么信息? 记 $\boldsymbol{Q}=[\boldsymbol{Q}_k, \boldsymbol{Q}_u]$，其中 $\boldsymbol{Q}_k=[\boldsymbol{q}_1,\cdots,\boldsymbol{q}_k]$ 而 $\boldsymbol{Q}_u=[\boldsymbol{q}_{k+1},\cdots,\boldsymbol{q}_n]$，注意我们仅仅算出 $\boldsymbol{Q}_k$ 和 $\boldsymbol{q}_{k+1}$；而 $\boldsymbol{Q}_u$ 的其余列是未知的，于是

$$\boldsymbol{H}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=[\boldsymbol{Q}_k,\boldsymbol{Q}_u]^{\mathrm{T}}\boldsymbol{A}[\boldsymbol{Q}_k,\boldsymbol{Q}_u]=\begin{bmatrix}\boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_k & \boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_u\\ \boldsymbol{Q}_u^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_k & \boldsymbol{Q}_u^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_u\end{bmatrix}$$

$$\begin{array}{cc} & \begin{array}{cc} k & n-k \end{array} \\ \equiv \begin{array}{c} k \\ n-k \end{array} & \begin{pmatrix}\boldsymbol{H}_k & \boldsymbol{H}_{uk}\\ \boldsymbol{H}_{ku} & \boldsymbol{H}_u\end{pmatrix}. \end{array} \tag{6.30}$$

注意因为 H 是上海森伯格阵，所以 H_k 是上海森伯格阵. 由于同样的理由，$\boldsymbol{H}_{ku}$ 在它的右上角(或许)有一个非零元 $h_{k+1,k}$. 因而 $\boldsymbol{H}_u$ 和 $\boldsymbol{H}_{uk}$ 是未知的；我们只知道 $\boldsymbol{H}_k$ 和 $\boldsymbol{H}_{ku}$.

303

当 $\boldsymbol{A}$ 对称时，$\boldsymbol{H}=\boldsymbol{T}$ 是对称和三对角的，并且使 Arnoldi 算法极大地简化，因为大多数 h_{ij} 为零：记

$$\boldsymbol{T}=\begin{bmatrix}\alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{n-1}\\ & & \beta_{n-1} & \alpha_n\end{bmatrix}.$$

由 $\boldsymbol{AQ}=\boldsymbol{QT}$ 的两边第 j 列相等得到

$$\boldsymbol{A}\boldsymbol{q}_j=\beta_{j-1}\boldsymbol{q}_{j-1}+\alpha_j\boldsymbol{q}_j+\beta_j\boldsymbol{q}_{j+1}.$$

因为 $\boldsymbol{Q}$ 的列是规范正交的，所以用 $\boldsymbol{q}_j^{\mathrm{T}}$ 乘这个等式的两边得到 $\boldsymbol{q}_j^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_j=\boldsymbol{\alpha}_j$. 这就证明下列阿诺尔迪算法的形式称为兰乔斯算法是恰当的.

算法 6.10 （部分）约化为对称三对角形式的兰乔斯算法

$q_1=\boldsymbol{b}/\|\boldsymbol{b}\|_2$，$\beta_0=0$，$q_0=0$
for $j=1$ to k
  $z=\boldsymbol{A}\boldsymbol{q}_j$
  $\alpha_j=\boldsymbol{q}_j^{\mathrm{T}}z$
  $z=z-\alpha_j\boldsymbol{q}_j-\beta_{j-1}\boldsymbol{q}_{j-1}$
  $\beta_j=\|z\|_2$
  if $\beta_j=0$，quit
  $\boldsymbol{q}_{j+1}=z/\beta_j$
end for

由兰乔斯算法计算的 q_j 通常称为兰乔斯向量. 在兰乔斯算法的第 k 步之后，下面是我们所获得的关于 $\boldsymbol{A}$ 的信息：

$$\begin{aligned}
\boldsymbol{T} &= \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = [\boldsymbol{Q}_k,\ \boldsymbol{Q}_u]^{\mathrm{T}}\boldsymbol{A}[\boldsymbol{Q}_k,\ \boldsymbol{Q}_u]^{\mathrm{T}} \\
&= \begin{bmatrix} \boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_k & \boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_u \\ \boldsymbol{Q}_u^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_k & \boldsymbol{Q}_u^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_u \end{bmatrix} \\
&\qquad\quad k \qquad n-k \\
&\equiv \begin{matrix} k \\ n-k \end{matrix} \begin{pmatrix} \boldsymbol{T}_k & \boldsymbol{T}_{uk} \\ \boldsymbol{T}_{ku} & \boldsymbol{T}_u \end{pmatrix} \\
&= \begin{bmatrix} \boldsymbol{T}_k & \boldsymbol{T}_{ku}^{\mathrm{T}} \\ \boldsymbol{T}_{ku} & \boldsymbol{T}_u \end{bmatrix}.
\end{aligned} \tag{6.31}$$

因为 $\boldsymbol{A}$ 对称，所以知道 $\boldsymbol{T}_k$ 对称并且 $\boldsymbol{T}_{ku} = \boldsymbol{T}_{uk}^{\mathrm{T}}$ 但不知道 $\boldsymbol{T}_u$. $\boldsymbol{T}_{ku}$ 在其右上角(或许)有一个非零元 β_k. 注意 β_k 非负，因为它作为 z 的范数被算出.

我们定义某些标准记号，它们对应于用阿诺尔迪算法和兰乔斯算法算出的 $\boldsymbol{A}$ 的部分分解.

304

定义 6.16 克雷洛夫子空间 $K_k(\boldsymbol{A},\boldsymbol{b})$ 是 $\mathrm{span}[\boldsymbol{b},\boldsymbol{A}\boldsymbol{b},\boldsymbol{A}^2\boldsymbol{b},\cdots,\boldsymbol{A}^{k-1}\boldsymbol{b}]$.

若根据上下文 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 是不言明的，则我们将用 K_k 代替 $K_k(\boldsymbol{A},\boldsymbol{b})$. 倘若算法不因为 $z=0$ 而停止，则由阿诺尔迪算法或兰乔斯算法算出的向量 Q_k 构成克雷洛夫子空间 K_k 的一个规范正交基.（可以证明 K_k 维数为 k 当且仅当阿诺尔迪算法或兰乔斯算法可以计算 $\boldsymbol{q}_k$ 而不先行停止；见问题6.12). 也称 H_k(或 T_k)为 $\boldsymbol{A}$ 在克雷洛夫子空间 K_k 上的投影.

我们的目标是设计只利用阿诺尔迪算法或兰乔斯算法的 k 步计算所得信息来求解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$. 希望 k 能比 n 小得多，因而算法有效.

(在第 7 章中将利用这个相同的信息求 $\boldsymbol{A}$ 的特征值. 我们已能概述如何做这件事情：注意若 $h_{k+1,k}$ 恰巧为零，则 $\boldsymbol{H}$(或 $\boldsymbol{T}$)是块上三角阵，因而 $\boldsymbol{H}_k$ 的所有特征值也是 $\boldsymbol{H}$ 的特征值，而且也是 $\boldsymbol{A}$ 的特征值，因为 $\boldsymbol{A}$ 和 $\boldsymbol{H}$ 相似. $\boldsymbol{H}_k$ 的(右)特征向量是 $\boldsymbol{H}$ 的特征向量，并且如果用 $\boldsymbol{Q}_k$ 乘它们，则得到 $\boldsymbol{A}$ 的特征向量. 当 $h_{k+1,k}$ 非零但很小时，预期 $\boldsymbol{H}_k$ 的特征值和特征向量提供 $\boldsymbol{A}$ 的特征值和特征向量的一个好的近似.)

通过指出舍入误差导致我们讨论的许多算法完全不同于这些算法按精确算术运算的结果来结束这个介绍. 特别地，由兰乔斯算法算出的向量 $\boldsymbol{q}_i$ 被很快地失去正交性，事实上它们通常变成线性相关. 这个明显极其糟糕的数值不稳定性导致研究人员在他们发现这个问题若干年之后放弃这些算法. 但最终研究人员认识到不是如何使算法稳定就是尽管其不稳定但还是收敛的！我们在 6.6.4 节中获得这些观点，那里我们分析求解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的共轭梯度法的收敛性(它是“不稳定的”但它总是收敛的)，在第 7 章中，特别在 7.4 节和 7.5 节中指出如何计算特征值(以及修正基本算法以保

证稳定性).

6.6.2　利用克雷洛夫子空间 K_k 解 $Ax=b$

仅仅给定来自阿诺尔迪算法或兰乔斯算法的第 k 步可利用的信息，我们如何求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 呢?

因为只知道向量是 $\boldsymbol{Q}_k$ 的列，所以“寻找”近似解的最合适的地方是在这些向量张成的克雷洛夫子空间 K_k 中. 换言之，观察下列形式“最佳的”逼近解.

$$\boldsymbol{x}_k=\sum_{j=1}^{k} z_k\boldsymbol{q}_k=\boldsymbol{Q}_k\cdot z\quad 其中 z=[z_1,\cdots,z_k]^{\mathrm{T}}.$$

305

现在必须定义“最佳的”. 有几种自然的但导致不同的算法的不同的定义. 设 $\boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{b}$ 表示真解而 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{Ax}_k$ 表示残差.

1. “最佳的” $\boldsymbol{x}_k$ 极小化 $\|\boldsymbol{x}_k-\boldsymbol{x}\|_2$. 遗憾地，在克雷洛夫子空间没有足够的信息去计算这个 $\boldsymbol{x}_k$.

2. “最佳的” $\boldsymbol{x}_k$ 极小化 $\|\boldsymbol{r}_k\|_2$. 这是可执行的，并且当 $\boldsymbol{A}$ 对称时相应的算法是 MINRES(极小残差)[194]，而当 $\boldsymbol{A}$ 非对称时相应的算法是 GMRES(广义极小残差)[215].

3. “最佳的” $\boldsymbol{x}_k$ 使 $\boldsymbol{r}_k\perp K_k$，即 $\boldsymbol{Q}_k^{\mathrm{T}}r_k=0$. 这有时称为*正交残差性质*，或类似有限元理论中相似的条件称为*伽辽金条件*. 当 A 对称时，相应的算法称为 SYMMLQ [194]. 当 $\boldsymbol{A}$ 非对称时，用一种变形的 GMRES 来工作.

4. 当 $\boldsymbol{A}$ 对称正定时，定义范数 $\|\boldsymbol{r}\|_{A^{-1}}=(\boldsymbol{r}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{r})^{1/2}$(见引理 1.3). 假定“最佳的” $\boldsymbol{x}_k$ 极小化 $\|\boldsymbol{r}_k\|_{A^{-1}}$. 这个范数与 $\|\boldsymbol{x}_k-\boldsymbol{x}\|_A$ 相同. 算法称为共轭梯度法[145].

当 A 对称正定时，“最佳的”最后两个定义还被证明是等价的.

定理 6.8　*设 $\boldsymbol{A}$ 对称，$\boldsymbol{T}_k=\boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{AQ}_k,\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{Ax}_k$，其中 $\boldsymbol{x}_k\in K_k$. 若 $\boldsymbol{T}_k$ 非奇异且 $\boldsymbol{x}_k=\boldsymbol{Q}_k\boldsymbol{T}_k^{-1}\boldsymbol{e}_1\|\boldsymbol{b}\|_2$，其中 $\boldsymbol{e}_1^{k\times 1}=[1,0,\cdots,0]^{\mathrm{T}}$，则 $\boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{r}_k=0$，若 $\boldsymbol{A}$ 还是正定的，则 $\boldsymbol{T}_k$ 必定非奇异，而且遍及一切 $x_k\in K_k$，这样选择的 $\boldsymbol{x}_k$ 也极小化 $\|\boldsymbol{r}_k\|_{A^{-1}}$. 还有 $\boldsymbol{r}_k=\pm\|\boldsymbol{r}_k\|_2\boldsymbol{q}_{k+1}$.*

证明　为简化记号去掉下标 k. 设 $x=\boldsymbol{QT}^{-1}\boldsymbol{e}_1\|\boldsymbol{b}\|_2$ 和 $\boldsymbol{r}=\boldsymbol{b}-\boldsymbol{Ax}$，并假定 $\boldsymbol{T}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{AQ}$ 非奇异. 通过计算确认 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{r}=0$.

$$\begin{aligned}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{r}=\boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{Ax})&=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}-\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Ax}\\&=\boldsymbol{e}_1\|\boldsymbol{b}\|_2-\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{QT}^{-1}\boldsymbol{e}_1\|\boldsymbol{b}\|_2)\\&\qquad\text{因为 } \boldsymbol{Q} \text{ 的第一列是 } \boldsymbol{b}/\|\boldsymbol{b}\|_2 \text{ 并且它的其他列正交于 } b\\&=\boldsymbol{e}_1\|\boldsymbol{b}\|_2-(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{AQ})\boldsymbol{T}^{-1}\boldsymbol{e}_1\|\boldsymbol{b}\|_2\\&=\boldsymbol{e}_1\|\boldsymbol{b}\|_2-(\boldsymbol{T})\boldsymbol{T}^{-1}\boldsymbol{e}_1\|\boldsymbol{b}\|_2\qquad\text{因为 } \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{AQ}=\boldsymbol{T}\\&=0.\end{aligned}$$

306

现在假定 $\boldsymbol{A}$ 也是正定的. 则 $\boldsymbol{T}$ 必须是正定的，因而也是非奇异的(见问题 6.13). 设 $\hat{\boldsymbol{x}}=\boldsymbol{x}+Qz$ 是 K 中另一候选的解，并设 $\hat{\boldsymbol{r}}=\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}$. 需要证明当 $z=0$ 时 $\|\hat{\boldsymbol{r}}\|_{A^{-1}}$

达到极小. 但是

$$
\begin{aligned}
\|\hat{r}\|_{A^{-1}}^2 &= \hat{r}^T A^{-1} \hat{r} \quad \text{由定义} \\
&= (r - AQz)^T A^{-1} (r - AQz) \\
&\qquad \text{因为} \hat{r} = b - A\hat{x} = b - A(x + Qz) = r - AQz \\
&= r^T A^{-1} r^T - 2(AQz)^T A^{-1} r + (AQz)^T A^{-1} (AQz) \\
&= \|r\|_{A^{-1}}^2 - 2z^T Q^T r + \|AQz\|_{A^{-1}}^2 \\
&\qquad \text{因为} (AQz)^T A^{-1} r = z^T Q^T A A^{-1} r = z^T Q^T r \\
&= \|r\|_{A^{-1}}^2 + \|AQz\|_{A^{-1}}^2 \quad \text{因为 } Q^T r = 0
\end{aligned}
$$

故 $\|\hat{r}\|_{A^{-1}}$ 达到极小当且仅当 $AQz = 0$. 因为 A 非奇异而 Q 列满秩, 故 $AQz = 0$ 当且仅当 $z = 0$.

为证明 $r_k = \pm \|r_k\|_2 q_{k+1}$, 再次引入下标. 因为 $x_k \in K_k$, 所以必有 $r_k = b - Ax_k \in K_{k+1}$, 故 r_k 是 Q_{k+1} 的列的一个线性组合, 因为这些列张成 K_{k+1}. 但是因为 $Q_k^T r_k = 0$, 所以 Q_{k+1} 唯一不正交于 r_k 的列是 q_{k+1}. □

6.6.3 共轭梯度法

关于对称正定阵选择的算法是 CG. 定理 6.8 刻画由 CG 计算的解 x_k 的特征. 虽然 MINRES 可能似乎比 CG 更自然, 因为它极小化 $\|r_k\|_2$ 而不是 $\|r_k\|_{A^{-1}}$, 然而, 人们还是证实了实施 MINRES 需要更多的工作, 并且更容易影响数值不稳定, 因而通常产生精度不及 CG 的解答. 我们将看到 CG 有特别吸引力的性质, 就是在存储器中同时只保留 4 个向量而不是 k 个向量(q_1 到 q_k)就可以执行算法. 而且在内循环中的工作除矩阵 - 向量积外, 限于两个点积, 三个 "saxpy" 运算(一个向量的倍数加上另一个向量), 以及少数标量运算. 这是一个非常小的工作量和存储量.

现在推导 CG. 有几种方法去做这个工作. 我们将从兰乔斯算法(算法 6.14)出发, 由定理 6.8 计算正交阵 Q_k 的列和三对角阵 T_k 的元素连同公式 $x_k = Q_k T_k^{-1} e_1 \|b\|_2$. 将指出如何经三个向量组的递推直接地计算 x_k. 同时存储器中只保留每个向量组的最新向量, 覆盖老的向量. 第一个向量组是近似解 x_k. 第二个向量组是残差 $r_k = b - Ax_k$, 定理 6.8 指出它平行于兰乔斯向量 q_{k+1}. 第三个向量组是共轭梯度 p_k. p_k 称为梯度是因为 CG 的一个单步可解释为选择一个标量 υ 使新的解 $x_k = x_{k-1} + \upsilon p_k$ 极小化残差范数 $\|r_k\|_{A^{-1}} = (r_k^T A^{-1} r_k)^{1/2}$. 换言之, 用 p_k 作为梯度搜索方向. p_k 称为共轭或更精确地称为 A - 共轭的是因为 $j \neq k$ 时 $p_k^T A p_j = 0$. 换言之, p_k 关于用 A 定义的内积是正交的(见引理 1.3). [307]

因为 A 对称正定, 故 $T_k = Q_k^T A Q_k$ 也是对称正定的. (见问题 6.13). 这意味着可对 T_k 执行楚列斯基分解得到 $T_k = \hat{L}_k \hat{L}_k^T = L_k D_k L_k^T$, 其中 L_k 是单位下双对角阵而 D_k 是对角阵. 然后由定理 6.8 利用 x_k 的公式得到

$$\begin{aligned} x_k &= Q_k T_k^{-1} e_1 \|b\|_2 \\ &= Q_k (L_k^{-T} D_k^{-1} L_k^{-1}) e_1 \|b\|_2 \\ &= (Q_k L_k^{-T})(D_k^{-1} L_k^{-1} e_1 \|b\|_2) \\ &\equiv (\widetilde{P}_k)(y_k), \end{aligned}$$

其中 $\widetilde{P}_k \equiv Q_k L_k^{-T}$ 和 $y_k \equiv D_k^{-1} L_k^{-1} e_1 \|b\|_2$. 记 $\widetilde{P}_k = [\widetilde{p}_1, \cdots, \widetilde{p}_k]$. 将证实共轭梯度 p_i 并行于 $\widetilde{P}_k$的列 $\widetilde{p}_i$. 我们证明下列引理就足够了.

引理 6.8　$\widetilde{P}_k$的列 $\widetilde{p}_i$是 A－共轭的. 换言之. $\widetilde{P}_k^{\mathrm{T}} A \widetilde{P}_k$是对角阵.

证明　计算

$$\begin{aligned} \widetilde{P}_k^{\mathrm{T}} A \widetilde{P}_k &= (Q_k L_k^{-T})^{\mathrm{T}} A (Q_k L_k^{-T}) = L_k^{-1}(Q_k^{\mathrm{T}} A Q_k) L_k^{-T} = L_k^{-1}(T_k) L_k^{-T} \\ &= L_k^{-1}(L_k D_k L_k^{\mathrm{T}}) L_k^{-T} = D_k. \end{aligned}$$
□

现在导出 $\widetilde{P}_k$的列和 y_k 的元素简单的递归. 我们将证明 $y_{k-1} \equiv [\eta_1, \cdots, \eta_{k-1}]^{\mathrm{T}}$ 与 $y_k = [\eta_1, \cdots, \eta_{k-1}, \eta_k]^{\mathrm{T}}$ 的前 $k-1$ 个元素完全相同以及 $\widetilde{P}_{k-1}$与 $\widetilde{P}_k$的前 $k-1$ 列完全相同. 所以可设

$$x_k = \widetilde{P}_k \cdot y_k = [\widetilde{P}_{k-1}, \widetilde{p}_k] \begin{bmatrix} y_{k-1} \\ \eta_k \end{bmatrix} = \widetilde{P}_{k-1} y_{k-1} + \widetilde{p}_k \eta_k = x_{k-1} + \widetilde{p}_k \eta_k \tag{6.32}$$

是关于 x_k 的递归.

关于 η_k的递归推导如下. 因为 T_{k-1}是 T_k 的前$(k-1)\times(k-1)$阶子阵, L_{k-1}和 D_{k-1}也分别是 L_k 和 D_k 的前$(k-1)\times(k-1)$阶子阵:

$$\begin{aligned} T_k &= \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{bmatrix} \\ &= L_k D_k L_k^{\mathrm{T}} \\ &= \begin{bmatrix} 1 & & & \\ l_1 & \ddots & & \\ & \ddots & \ddots & \\ & & l_{k-1} & 1 \end{bmatrix} \cdot \begin{bmatrix} d_1 & & & \\ & \ddots & & \\ & & d_{k-1} & \\ & & & d_k \end{bmatrix} \cdot \begin{bmatrix} 1 & & & \\ l_1 & \ddots & & \\ & \ddots & \ddots & \\ & & l_{k-1} & 1 \end{bmatrix}^{\mathrm{T}} \\ &= \begin{bmatrix} L_{k-1} & \\ l_{k-1}\hat{e}_{k-1}^{\mathrm{T}} & 1 \end{bmatrix} . \ \mathrm{diag}\,(D_{k-1},\ d_k). \begin{bmatrix} L_{k-1} & \\ l_{k-1}\hat{e}_{k-1}^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}}, \end{aligned}$$

308

其中$\hat{e}_{k-1}^{\mathrm{T}} = [0, \cdots, 0, 1]$维数为 $k-1$. 类似地, D_{k-1}^{-1}和 L_{k-1}^{-1}分别是 $D_k^{-1} = \mathrm{diag}(D_{k-1}^{-1}, \mathrm{d}_k^{-1})$和

$$L_k^{-1} = \begin{bmatrix} L_{k-1}^{-1} & \\ \star & 1 \end{bmatrix},$$

的前$(k-1)\times(k-1)$阶子阵，其中最后行$\star$的细节我们不关心. 这意味着$\boldsymbol{y}_{k-1}=\boldsymbol{D}_{k-1}^{-1}\boldsymbol{L}_{k-1}^{-1}\hat{\boldsymbol{e}}_1\|\boldsymbol{b}\|_2$（其中$\hat{\boldsymbol{e}}_1$维数为$k-1$）与

$$\boldsymbol{y}_k=\boldsymbol{D}_k^{-1}\boldsymbol{L}_k^{-1}\boldsymbol{e}_1\|\boldsymbol{b}\|_2=\begin{bmatrix}\boldsymbol{D}_{k-1}^{-1} & \\ & d_k^{-1}\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{L}_{k-1}^{-1} & \\ \star & 1\end{bmatrix}\cdot\boldsymbol{e}_1\|\boldsymbol{b}\|_2=\begin{bmatrix}\boldsymbol{D}_{k-1}^{-1}\boldsymbol{L}_{k-1}^{-1}\hat{\boldsymbol{e}}_1\|\boldsymbol{b}\|_2\\ \eta_k\end{bmatrix}=\begin{bmatrix}\boldsymbol{y}_{k-1}\\ \eta_k\end{bmatrix}$$

的前$k-1$个分量完全相同.

现在需要一个关于$\widetilde{\boldsymbol{P}}_k=[\tilde{\boldsymbol{p}}_1,\cdots,\tilde{\boldsymbol{p}}_k]$的列的递归. 因为$\boldsymbol{L}_{k-1}^{\mathrm{T}}$是上三角阵，故$\boldsymbol{L}_{k-1}^{-\mathrm{T}}$也是上三角阵，它构成$\boldsymbol{L}_k^{-\mathrm{T}}$的前$(k-1)\times(k-1)$阶子阵. 所以$\widetilde{\boldsymbol{P}}_{k-1}$与

$$\widetilde{\boldsymbol{P}}_k=\boldsymbol{Q}_k\boldsymbol{L}_k^{-\mathrm{T}}=[\boldsymbol{Q}_{k-1},\boldsymbol{q}_k]\begin{bmatrix}\boldsymbol{L}_{k-1}^{-\mathrm{T}} & \star\\ 0 & 1\end{bmatrix}=[\boldsymbol{Q}_{k-1}\boldsymbol{L}_{k-1}^{-\mathrm{T}},\tilde{\boldsymbol{p}}_k]=[\widetilde{\boldsymbol{P}}_{k-1},\tilde{\boldsymbol{p}}_k]$$

的前$k-1$列完全相同. 从$\widetilde{\boldsymbol{P}}_k=\boldsymbol{Q}_k\boldsymbol{L}_k^{-\mathrm{T}}$得到$\widetilde{\boldsymbol{P}}_k\boldsymbol{L}_k^{\mathrm{T}}=\boldsymbol{Q}_k$，由此式两边第$k$列相等得到递归

$$\tilde{\boldsymbol{p}}_k=\boldsymbol{q}_k-\boldsymbol{l}_{k-1}\tilde{\boldsymbol{p}}_{k-1}. \tag{6.33}$$

总之，我们有$\boldsymbol{q}_k$（从兰乔斯算法），$\tilde{\boldsymbol{p}}_k$（从(6.33)式），和近似解$\boldsymbol{x}_k$（从(6.32)式）的递归. 所有这些递归是简短的，即，为了执行它们只需要前面或前面两个迭代. 因而当存储少量的向量并在内循环中做少量的点积、saxpys 和标量运算时，它们合在一起提供计算$\boldsymbol{x}_k$的方法.

为得到最终的 CG 算法还必须稍微地简化这些递归. 因为定理6.8告诉我们$\boldsymbol{r}_k$和$\boldsymbol{q}_{k+1}$并行，所以可用递归$\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$或等价地$\boldsymbol{r}_k=\boldsymbol{r}_{k-1}-\boldsymbol{\eta}_k\boldsymbol{A}\tilde{\boldsymbol{p}}_k$（用$\boldsymbol{A}$乘递归$\boldsymbol{x}_k=\boldsymbol{x}_{k-1}+\boldsymbol{\eta}_k\tilde{\boldsymbol{p}}_k$并减去$\boldsymbol{b}=\boldsymbol{b}$得到）替代$\boldsymbol{q}_{k+1}$的兰乔斯递归. 这样得到三个向量递归. 309

$$\boldsymbol{r}_k=\boldsymbol{r}_{k-1}-\boldsymbol{\eta}_k\boldsymbol{A}\tilde{\boldsymbol{p}}_k, \tag{6.34}$$

$$\boldsymbol{x}_k=\boldsymbol{x}_{k-1}+\boldsymbol{\eta}_k\tilde{\boldsymbol{p}}_k \qquad \text{从(6.32)式}, \tag{6.35}$$

$$\tilde{\boldsymbol{p}}_k=\boldsymbol{q}_k-\boldsymbol{l}_{k-1}\tilde{\boldsymbol{p}}_{k-1} \qquad \text{从(6.33)式}. \tag{6.36}$$

为了消去$\boldsymbol{q}_k$，把$\boldsymbol{q}_k=\boldsymbol{r}_{k-1}/\|\boldsymbol{r}_{k-1}\|_2$和$\boldsymbol{p}_k\equiv\|\boldsymbol{r}_{k-1}\|_2\tilde{\boldsymbol{p}}_k$代入到上面的递归得到

$$\begin{aligned}\boldsymbol{r}_k&=\boldsymbol{r}_{k-1}-\frac{\boldsymbol{\eta}_k}{\|\boldsymbol{r}_{k-1}\|_2}\boldsymbol{A}\boldsymbol{p}_k\\&\equiv\boldsymbol{r}_{k-1}-\upsilon_k\boldsymbol{A}\boldsymbol{p}_k,\end{aligned} \tag{6.37}$$

$$\begin{aligned}\boldsymbol{x}_k&=\boldsymbol{x}_{k-1}+\frac{\boldsymbol{\eta}_k}{\|\boldsymbol{r}_{k-1}\|_2}\boldsymbol{p}_k\\&\equiv\boldsymbol{x}_{k-1}+\upsilon_k\boldsymbol{p}_k,\end{aligned} \tag{6.38}$$

$$\begin{aligned}\boldsymbol{p}_k&=\boldsymbol{r}_{k-1}-\frac{\|\boldsymbol{r}_{k-1}\|_2\boldsymbol{l}_{k-1}}{\|\boldsymbol{r}_{k-2}\|_2}\boldsymbol{p}_{k-1}\\&\equiv\boldsymbol{r}_{k-1}+\mu_k\cdot\boldsymbol{p}_{k-1}.\end{aligned} \tag{6.39}$$

我们仍然需要标量υ_k和μ_k的公式. 正如将看到的那样，根据算法算出的向量的

点积它们有好几个等价的数学表达式. 最终的公式是选择极小化所需的点积的次数，并且因为它们比其他的选择更稳定.

为得到 v_k 的公式，首先用 $\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}$ 左乘(6.39)式两边，并利用 $\boldsymbol{p}_k$ 和 $\boldsymbol{p}_{k-1}$ 是 $\boldsymbol{A}$ - 共轭的事实(引理6.8)得到

$$\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k = \boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{r}_{k-1} + 0 = \boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k. \tag{6.40}$$

然后，用 $\boldsymbol{r}_{k-1}^{T}$ 左乘(6.37)式两边并用 $\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_k = 0$ 的事实(因为 $\boldsymbol{r}_i$ 平行于正交阵 $\boldsymbol{Q}$ 的列)得到

$$v_k = \frac{\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1}}{\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k} = \frac{\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1}}{\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k} \quad \text{由(6.40)式} \tag{6.41}$$

也能从定理6.8中 v_k 的性质导出(6.41)式，即它使残差范数达到极小

$$\begin{aligned}\|\boldsymbol{r}_k\|_{A^{-1}}^2 &= \boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{r}_k \\ &= (\boldsymbol{r}_{k-1} - v_k\boldsymbol{A}\boldsymbol{p}_k)^{\mathrm{T}}\boldsymbol{A}^{-1}(\boldsymbol{r}_{k-1} - v_k\boldsymbol{A}\boldsymbol{p}_k) \qquad \text{由(6.37)式} \\ &= \boldsymbol{r}_{k-1}\boldsymbol{A}^{-1}\boldsymbol{r}_{k-1}^{\mathrm{T}} - 2v_k\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{r}_{k-1} + v_k^2\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k.\end{aligned}$$

[310]

这个表达式是 v_k 的二次函数，故通过置它的关于 v_k 的导数为零并求解 v_k 可容易地使它达到极小. 这样得到

$$\begin{aligned} v_k &= \frac{\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{r}_{k-1}}{\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k} \\ &= \frac{(\boldsymbol{r}_{k-1} + \boldsymbol{\mu}_k \cdot \boldsymbol{p}_{k-1})^{\mathrm{T}}\boldsymbol{r}_{k-1}}{\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k} \quad \text{由(6.39)式} \\ &= \frac{\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1}}{\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k}, \end{aligned}$$

上面利用了 $\boldsymbol{p}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1} = 0$ 的事实，因为 $\boldsymbol{r}_{k-1}$ 正交于 K_{k-1} 中包括 $\boldsymbol{p}_{k-1}$ 的所有向量，所以 $\boldsymbol{p}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1} = 0$ 成立.)

为了得到 μ_k 的公式，用 $\boldsymbol{p}_{k-1}^{\mathrm{T}}\boldsymbol{A}$ 左乘(6.39)式的两边，并利用 p_k 和 p_{k-1} 是 A - 共轭的事实(引理6.8)得到

$$\mu_k = -\frac{\boldsymbol{p}_{k-1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{r}_{k-1}}{\boldsymbol{p}_{k-1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_{k-1}} \tag{6.42}$$

这个公式的不足是它除了 v_k 需要的两个点积外还需要另一个点积 $p_{k-1}^{\mathrm{T}}Ar_{k-1}$. 因而我们将导出不需要新的点积的另一个公式.

通过导出 v_k 的另一个公式来做这件事：用 $\boldsymbol{r}_k^{\mathrm{T}}$ 左乘(6.37)式两边，再利用 $\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_k = 0$ 并求解 v_k 得到

$$v_k = -\frac{\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k}{\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k}. \tag{6.43}$$

关于 v_{k-1} 的两个表达式(6.41)和(6.43)相等(注意，已从下标中减去1)，重新整理并与(6.42)比较得到 μ_k 最终的公式：

$$\mu_k = -\frac{\boldsymbol{p}_{k-1}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{r}_{k-1}}{\boldsymbol{p}_{k-1}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{p}_{k-1}} = \frac{\boldsymbol{r}_{k-1}^{\mathrm{T}} \boldsymbol{r}_{k-1}}{\boldsymbol{r}_{k-2}^{\mathrm{T}} \boldsymbol{r}_{k-2}}. \tag{6.44}$$

合并递归式(6.37)，(6.38)和(6.39)以及公式(6.41)，(6.44)得到共轭梯度法的最终实施过程.

算法 6.11 共轭梯度法：

$k=0; x_0=0; \boldsymbol{r}_0=\boldsymbol{b}; \boldsymbol{p}_1=\boldsymbol{b};$

repeat

  $k=k+1$

  $z=\boldsymbol{A}\cdot\boldsymbol{p}_k$

  $\upsilon_k=(\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1})/(\boldsymbol{p}_k^{\mathrm{T}}z)$

  $\boldsymbol{x}_k=\boldsymbol{x}_{k-1}+\upsilon_k\boldsymbol{p}_k$

  $\boldsymbol{r}_k=\boldsymbol{r}_{k-1}-\upsilon_k z$

  $\mu_{k+1}=(\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k)/(\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1})$

  $\boldsymbol{p}_{k+1}=\boldsymbol{r}_k+\mu_{k+1}\boldsymbol{p}_k$

until $\|\boldsymbol{r}_k\|_2$ 足够小

311

CG 内循环的花费是一次矩阵 - 向量积 $z=\boldsymbol{A}\cdot\boldsymbol{p}_k$，两个内积(从一次循环迭代到下一次迭代保存 $\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k$ 的值)，三次 saxpys 和少数标量运算. 需要存储的向量仅仅是 $\boldsymbol{r}$，$\boldsymbol{x}$，$\boldsymbol{p}$ 和 $z=\boldsymbol{A}\boldsymbol{p}$ 的最新的值. 执行过程更多的细节，包括如何判定"$\|\boldsymbol{r}_k\|_2$ 足够小"，见 NETLIB/templates/Templates.html.

6.6.4 共轭梯度法的收敛性分析

从仅依赖于 $\boldsymbol{A}$ 的条件数的 CG 收敛性分析开始. 这个分析将证明以一个小于 1 的固定的倍数减小误差所需的 CG 迭代次数与条件数的平方根成比例. 这个最差情况分析对模型问题泊松方程的收敛速度是一个好的估计. 但对许多其他情况它严重地低估收敛速度. 提出基于条件数的界之后，我们将描述何时能期望较快收敛.

从初始近似解 $\boldsymbol{x}_0=0$ 入手. 记得取遍一切可能的解 $\boldsymbol{x}_k\in K_k(\boldsymbol{A},\boldsymbol{b})$，$\boldsymbol{x}_k$ 使残差 $r_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$ 的 $\boldsymbol{A}^{-1}$ - 范数达到极小. 这意味着取遍一切 $z\in K_k=\mathrm{span}[\boldsymbol{b},\boldsymbol{A}\boldsymbol{b},\boldsymbol{A}^2\boldsymbol{b},\cdots,\boldsymbol{A}^{k-1}\boldsymbol{b}]$，$x_k$ 使

$$\|\boldsymbol{b}-\boldsymbol{A}z\|_{A^{-1}}^2\equiv f(z)=(\boldsymbol{b}-\boldsymbol{A}z)^{\mathrm{T}}\boldsymbol{A}^{-1}(\boldsymbol{b}-\boldsymbol{A}z)=(\boldsymbol{x}-z)^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{x}-z)$$

达到极小. 任意的 $z\in K_k(A,b)$ 可写成 $z=\sum_{j=0}^{k-1}\alpha_j\boldsymbol{A}^j\boldsymbol{b}=\boldsymbol{p}_{k-1}(\boldsymbol{A})\boldsymbol{b}=\boldsymbol{p}_{k-1}(\boldsymbol{A})\boldsymbol{A}\boldsymbol{x}$，其中 $p_{n-1}(\xi)=\sum_{j=0}^{k-1}\alpha_j\xi^j$ 是 $k-1$ 次多项式. 所以

$$\begin{aligned} f(z) &= [(\boldsymbol{I}-p_{k-1}(\boldsymbol{A})\boldsymbol{A})\boldsymbol{x}]^{\mathrm{T}}\boldsymbol{A}[(\boldsymbol{I}-p_{k-1}(\boldsymbol{A})\boldsymbol{A})\boldsymbol{x}] \\ &\equiv (\boldsymbol{q}_k(\boldsymbol{A})\boldsymbol{x})^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{q}_k(\boldsymbol{A})\boldsymbol{x}) \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{q}_k(\boldsymbol{A})\boldsymbol{A}\boldsymbol{q}_k(\boldsymbol{A})\boldsymbol{x}, \end{aligned}$$

其中 $q_k(\xi)\equiv 1-p_{k-1}(\xi)\cdot\xi$ 是 $q_k(0)=1$ 的 k 次多项式. 注意因为 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$，所以 $(q_k(\boldsymbol{A}))^{\mathrm{T}}=q_k(\boldsymbol{A})$. 设$\mathcal{Q}_k$ 是在0上取值为1的所有 k 次多项式的集合，这意味着

312

$$f(x_k)=\min_{z\in K_k} f(z)=\min_{q_k\in\mathcal{Q}_k}\boldsymbol{x}^{\mathrm{T}}q_k(\boldsymbol{A})\boldsymbol{A}q_k(\boldsymbol{A})\boldsymbol{x}. \tag{6.45}$$

为简化这个表达式，记特征分解 $\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}}$ 并设 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{y}$ 所以

$$\begin{aligned} f(x_k) = \min_{z\in K_k} f(z) &= \min_{q_k\in\mathcal{Q}_k}\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{q}_k(\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}}))(\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}})(\boldsymbol{q}_k(\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}}))\boldsymbol{x} \\ &= \min_{q_k\in\mathcal{Q}_k}\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{Q}\boldsymbol{q}_k(\boldsymbol{\Lambda})\boldsymbol{Q}^{\mathrm{T}})(\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q})^{\mathrm{T}}(\boldsymbol{Q}\boldsymbol{q}_k(\boldsymbol{\Lambda})\boldsymbol{Q}^{\mathrm{T}})\boldsymbol{x} \\ &= \min_{q_k\in\mathcal{Q}_k}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{q}_k(\boldsymbol{\Lambda})\boldsymbol{\Lambda}\boldsymbol{q}_k(\boldsymbol{\Lambda})\boldsymbol{y} \\ &= \min_{q_k\in\mathcal{Q}_k}\boldsymbol{y}^{\mathrm{T}}\cdot\mathrm{diag}(\boldsymbol{q}_k(\boldsymbol{\lambda}_i)\boldsymbol{\lambda}_i\boldsymbol{q}_k(\boldsymbol{\lambda}_i))\cdot\boldsymbol{y} \\ &= \min_{q_k\in\mathcal{Q}_k}\sum_{i=1}^{n}\boldsymbol{y}_i^2\lambda_i(\boldsymbol{q}_k(\boldsymbol{\lambda}_i))^2 \\ &\leqslant \min_{q_k\in\mathcal{Q}_k}\left(\max_{\lambda_i\in\lambda(A)}(\boldsymbol{q}_k(\boldsymbol{\lambda}_i))^2\right)\sum_{i=1}^{n}y_i^2\lambda_i \\ &= \min_{q_k\in\mathcal{Q}_k}\left(\max_{\lambda_i\in\lambda(A)}(q_k(\lambda_i))^2\right)f(x_0) \end{aligned}$$

因为 $x_0=0$ 推出 $f(x_0)=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{y}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{y}=\sum_{i=1}^{n}y_i^2\lambda_i$. 所以

$$\frac{\|r_k\|_{A^{-1}}^2}{\|r_0\|_{A^{-1}}^2}=\frac{f(x_k)}{f(x_0)}\leqslant\min_{q_k\in\mathcal{Q}}\ \max_{\lambda_i\in\lambda(A)}(q_k(\lambda_i))^2$$

或

$$\frac{\|r_k\|_{A^{-1}}}{\|r_0\|_{A^{-1}}}\leqslant\min_{q_k\in\mathcal{Q}}\ \max_{\lambda_i\in\lambda(A)}|q_k(\lambda_i)|$$

因此已把CG收敛多快的问题转化为关于多项式的问题：当 ξ 在 $\boldsymbol{A}$ 的特征值的范围内变化同时满足 $q_k(0)=1$ 的 k 次多项式 $q_k(\xi)$ 能多小？因为 $\boldsymbol{A}$ 正定，其特征值位于区间$[\lambda_{\min},\lambda_{\max}]$中，其中 $0<\lambda_{\min}\leqslant\lambda_{\max}$，故为了得到一个较简单的上界，我们将换成寻找一个在整个区间$[\lambda_{\min},\lambda_{\max}]$是小的而且在0上的值为1的 k 次多项式 $\hat{q}_k(\xi)$. 从6.5.6节中讨论的切比雪夫多项式 $T_k(\xi)$ 容易构造具有这个性质的多项式 $\hat{q}_k(\xi)$. 记得当$|\xi|\leqslant 1$ 时$|T_k(\xi)|\leqslant 1$ 并且当$|\xi|>1$ 时迅速递增(见图6-6). 现在设

$$\hat{q}_k(\xi)=\boldsymbol{T}_k\left(\frac{\lambda_{\max}+\lambda_{\min}-2\xi}{\lambda_{\max}-\lambda_{\min}}\right)\Big/\boldsymbol{T}_k\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right).$$

容易看出$\hat{q}_k(0)=1$，并且若 $\xi\in[\lambda_{\min},\lambda_{\max}]$，则

$$\left|\frac{\lambda_{\max}+\lambda_{\min}-2\xi}{\lambda_{\max}-\lambda_{\min}}\right|\leqslant 1,$$

故

$$\frac{\|\boldsymbol{r}_k\|_{A^{-1}}}{\|\boldsymbol{r}_0\|_{A^{-1}}}\leqslant\min_{q_k\in Q}\max_{\lambda_i\in\lambda(A)}|q_k(\lambda_i)|$$

$$\leqslant\frac{1}{T_k\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right)}=\frac{1}{T_k\left(\frac{\kappa+1}{\kappa-1}\right)}=\frac{1}{T_k\left(1+\frac{2}{\kappa-1}\right)},\tag{6.46}$$

313

其中 $\kappa=\lambda_{\max}/\lambda_{\min}$ 是 A 的条件数.

若条件数 κ 接近于 1，则 $1+2/(\kappa-1)$ 大，$1/T_k\left(1+\frac{2}{\kappa-1}\right)$ 小，并且收敛迅速. 若 κ 大，则收敛慢下来，残差 r_k 的 $\boldsymbol{A}^{-1}$ - 范数像

$$\frac{1}{T_k\left(1+\frac{2}{\kappa-1}\right)}\leqslant\frac{2}{1+\frac{2\kappa}{\sqrt{\kappa-1}}}$$

一样地趋于零.

例 6.14 对 $N\times N$ 模型问题，$\kappa=O(N^2)$. 因而 CG 的 k 步之后残差大约以 $(1-O(N^{-1}))^k$ 相乘，与具有最佳松弛参数 ω 的 SOR 法相同. 换言之，CG 作 $O(N)=O(n^{1/2})$ 次迭代收敛. 因为每次迭代花费 $O(n)$，所以全部的花费为 $O(n^{3/2})$. 这就说明表 6-1 中关于 CG 的表列数据. ◇

这个利用条件数的分析不能说明 CG 的全部重要的收敛性态. 下例说明 $\boldsymbol{A}$ 的特征值整体的分布是重要的而不单是最大的和最小的特征值之比.

例 6.15 考虑图 6-7，它对 8 种不同的线性方程组在每个 CG 步上画出相对残差 $\|r_k\|_2/\|r_0\|_2$. 相对残差度量收敛速度. 当首次出现这个比降低到 10^{-13} 以下时或 $k=200$ 步之后，CG 执行过程终止.

所示的所有 8 个线性方程组有相同的维数 $n=10^4$ 和相同的条件数 $\kappa\approx4134$，然而它们的收敛性态根本不同. 最上面的(虚)线是 $1/T_k\left(1+\frac{2}{\kappa-1}\right)$，不等式(6.46)告诉我们它是 $\|r_k\|_{A^{-1}}/\|r_0\|_{A^{-1}}$ 的一个上界. 它证实 $\|r_k\|_2/\|r_0\|_2$ 的图像和 $\|r_k\|_{A^{-1}}/\|r_0\|_{A^{-1}}$ 的图像几乎相同，所以我们只画出前者的图像，这些是较容易说明的.

实线是关于 100×100 网格点上具有一个随机的右端项 b 的泊松方程的 $\|r_k\|_2/\|r_0\|_2$. 观察到上界保存它的一般的收敛性态. 从左边的 D_1 到右边的 D_7 编号的七条长划线是七个对角线性方程组 $\boldsymbol{D}_ix=b$ 的 $\|r_k\|_2/\|r_0\|_2$ 的图像. 每个 D_i 有相同的维数并且有像泊松方程一样的条件数，因而为了了解它们不同的收敛性态需要更仔细地研究它们.

我们已构造每个 $\boldsymbol{D}_i$ 使它的最小的 m_i 个特征值和最大的 m_i 个特征值与泊松方程的那些特征值完全相同，其余的 $n-2m_i$ 个特征值等于最大的特征值和最小的特征值的几何平均. 换言之，D_i 只有 $d_i=2m_i+1$ 个不同的特征值. 设 k_i 表示 $D_ix=b$ 的解达到 $\|r_k\|_2/\|r_0\|_2\leqslant10^{-13}$ 所需要的 CG 迭代次数. 收敛性质总结在下表中：

例题编号	i	1	2	3	4	5	6	7
不同的特征值个数	d_i	3	11	41	81	201	401	5000
收敛所需的步数	k_i	3	11	27	59	94	134	>200

观察到收敛所需的步数 k_i 随不同的特征值个数 d_i 增长. D_7 有像泊松方程一样的相同的谱并且缓慢地收敛.

在不出现舍入时，我们断言 CG 收敛应该精确地作 $k_i = d_i$ 步. 理由是可找一个 d_i 次的多项式 $q_{d_i}(\xi)$，在 $\boldsymbol{A}$ 的特征值 α_j 上取值为 0，另一方面 $q_{d_i}(0) = 1$，即

$$q_{d_i}(\xi) = \frac{\prod_{j=1}^{d_i}(\alpha_j - \xi)}{\prod_{j=1}^{d_i}(\alpha_j)}.$$

(6.45)式告诉我们 d_i 步之后，取遍一切可能的在 0 上取值为 1 的 d_i 次多项式 CG 极小化 $\|r_{d_i}\|_{A^{-1}}^2 = f(x_{d_i})$. 因为 q_{d_i} 是那些多项式之一并且 $q_{d_i}(\boldsymbol{A}) = 0$. 所以必有 $\|r_{d_i}\|_{A^{-1}}^2 = 0$ 或 $r_{d_i} = 0$. ◇

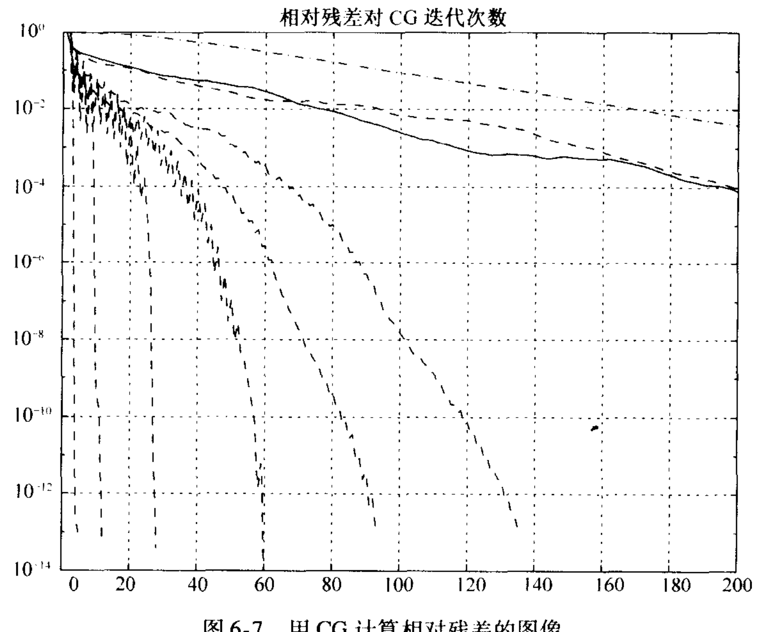


图6-7　用CG计算相对残差的图像

314 ~ 315

例6.15的一个经验是若 $\boldsymbol{A}$ 的最大的和最小的特征值个数总共很少(或成串地靠近在一起)，则 CG 比只基于 $\boldsymbol{A}$ 的条件数指出的分析更加快速地收敛.

另一个经验是按浮点的算术运算的 CG 的性态与它按精确的算术运算的性态可能相当大的不同. 我们观察到这是因为不同的特征值的个数 d_i 经常地不同于收敛所需要的步数 k_i，虽然理论上我们证明它们应该相同. 尽管如此，d_i 和 k_i 还是具有相同的数量级.

事实上，如果按精确的算术运算执行 CG 并且与按浮点算术运算计算的那些计算解和残差作比较，后者非常可能发散，马上看出是彻底相异的. 尽管如此，只要 A 不太病态，浮点运算的结果最终将收敛于 $Ax=b$ 所要求的解，因而 CG 还是非常有用的. 精确的和浮点的结果可能显著不同. 这一事实是有趣的但是并不妨碍 CG 的实际使用.

当发现 CG 时，它被证明按精确的算术运算，它在 n 步之后将提供精确解，因为 r_{n+1} 应正交于 n 个其他的正交的向量 r_1 到 r_n，因而 r_{n+1} 必为零. 换言之，CG 被认为是一个直接法而不是迭代法. 实际上当 n 步之后收敛不出现时，CG 被认为是不稳定的而被弃用了许多年. 最终它被认为是一个相当好的迭代法，通常在 $k \ll n$ 步之后提供十分精确的解答.

最近，设计了一个巧妙的向后误差分析去说明 CG 按浮点计算所表现出来的性态，并说明它可以如何不同于精确的算术运算[123]. 这个性态也可包括收敛中长久的“平稳时期”并且对许多次迭代穿插一些快速收敛的周期，$\|r_k\|_2$ 仅递减一点点. 这个性态可以说明如下：证明 CG 按浮点算术运算应用于 $Ax=b$ 表现恰好像 CG 按精确的算术运算应用于 $\tilde{A}\tilde{x}=\tilde{b}$，其中 $\tilde{A}$ 在下列意义下接近 A：$\tilde{A}$ 比 A 有更大的维数，但 $\tilde{A}$ 的特征值全部位于围绕 A 的特征值附近狭小的串中. 因而收敛中的平稳时期对应于构成 CG 的多项式 q_k 越来越多的零点接近于 $\tilde{A}$ 位于一个串中的特征值.

6.6.5 预条件

在前节中看到 CG 的收敛率依赖于 A 的条件数或者更一般地依赖于 A 的特征值的分布. 其他的克雷洛夫子空间方法具有同样的性质. 预条件意味着用方程组 $M^{-1}Ax=M^{-1}b$ 代替 $Ax=b$，其中 M 是 A 的一个近似，具有下列性质

1. M 对称正定；
2. $M^{-1}A$ 良态或有少数极端的特征值；
3. $Mx=b$ 容易求解.

一种仔细的随问题而定的 M 的选择通常能使 $M^{-1}A$ 的条件数比 A 的条件数小，从而显著地加速收敛. 实际上，对迭代法不管怎样收敛一个好的预条件子通常是必要的，迭代法中当前的许多研究集中在寻找较好的预条件子(也见 6.10 节). [316]

因为 $M^{-1}A$ 一般不对称，所以不能直接对方程组 $M^{-1}Ax=M^{-1}b$ 应用 CG. 下面我们导出预条件共轭梯度法. 设 $M=Q\Lambda Q^{\mathrm{T}}$ 是 M 的特征分解，并定义 $M^{1/2}\equiv Q\Lambda^{1/2}Q^{\mathrm{T}}$. 注意 $M^{1/2}$ 也是对称正定的，且 $(M^{1/2})^2=M$. 现在用 $M^{1/2}$ 乘 $M^{-1}Ax=M^{-1}b$ 得到新的对称正定方程组 $(M^{-1/2}AM^{-1/2})(M^{1/2}x)=M^{-1/2}b$，或 $\hat{A}\hat{x}=\hat{b}$. 注意因为 $\hat{A}$ 和 $M^{-1}A$ 相似($M^{-1}A=M^{-1/2}\hat{A}M^{1/2}$)，所以它们有相同的特征值. 现在以一种避免需要用 $M^{-1/2}$ 相乘的方法对 $\hat{A}\hat{x}=\hat{b}$ 隐式地应用 CG. 这样得到下列算法.

算法6.12 预条件CG算法：

$k=0; x_0=0; r_0=b; p_1=M^{-1}b; y_0=M^{-1}r_0$

repeat

$k=k+1$

$z=\boldsymbol{A}\cdot\boldsymbol{p}_k$

$\upsilon_k=(\boldsymbol{y}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1})/(\boldsymbol{p}_k^{\mathrm{T}}z)$

$\boldsymbol{x}_k=\boldsymbol{x}_{k-1}+\upsilon_k\boldsymbol{p}_k$

$\boldsymbol{r}_k=\boldsymbol{r}_{k-1}-\upsilon_k z$

$\boldsymbol{y}_k=\boldsymbol{M}^{-1}\boldsymbol{r}_k$

$\boldsymbol{\mu}_{k+1}=(\boldsymbol{y}_k^{\mathrm{T}}\boldsymbol{r}_k)/(\boldsymbol{y}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1})$

$\boldsymbol{p}_{k+1}=\boldsymbol{y}_k+\mu_{k+1}\boldsymbol{p}_k$

until $\|\boldsymbol{r}_k\|_2$足够小

定理6.9 设$\boldsymbol{A}$和$\boldsymbol{M}$对称正定，$\hat{\boldsymbol{A}}=\boldsymbol{M}^{-1/2}\boldsymbol{A}\boldsymbol{M}^{-1/2}$，$\hat{\boldsymbol{b}}=\boldsymbol{M}^{-1/2}\hat{\boldsymbol{b}}$. CG算法应用于$\hat{\boldsymbol{A}}\hat{\boldsymbol{x}}=\hat{\boldsymbol{b}}$

$\boldsymbol{k}=0; \hat{\boldsymbol{x}}_0=0; \hat{\boldsymbol{r}}_0=\hat{\boldsymbol{b}};\ \hat{\boldsymbol{p}}_1=\hat{\boldsymbol{b}};$

repeat

$k=k+1$

$\hat{z}=\hat{A}\cdot\hat{\boldsymbol{p}}_k$

$\hat{\boldsymbol{\upsilon}}_k=(\hat{\boldsymbol{r}}_{k-1}^{\mathrm{T}}\hat{\boldsymbol{r}}_{k-1})/(\hat{\boldsymbol{p}}_k^{\mathrm{T}}\hat{z})$

$\hat{\boldsymbol{x}}_k=\hat{\boldsymbol{x}}_{k-1}+\hat{\upsilon}_k\hat{\boldsymbol{p}}_k$

$\hat{\boldsymbol{r}}_k=\hat{\boldsymbol{r}}_{k-1}-\hat{\upsilon}_k\hat{z}$

$\hat{\mu}_{k+1}=(\hat{\boldsymbol{r}}_k^{\mathrm{T}}\hat{\boldsymbol{r}}_k)/(\hat{\boldsymbol{r}}_{k-1}^{\mathrm{T}}\hat{\boldsymbol{r}}_{k-1})$

$\hat{\boldsymbol{p}}_{k+1}=\hat{\boldsymbol{r}}_k+\hat{\mu}_{k+1}\hat{\boldsymbol{p}}_k$

until $\|\hat{\boldsymbol{r}}_k\|_2$足够小

并且与算法6.12的关系如下：

317

$$\hat{\mu}_k=\mu_k,$$
$$\hat{\upsilon}_k=\upsilon_k,$$
$$\hat{z}=\boldsymbol{M}^{-1/2}z,$$
$$\hat{\boldsymbol{x}}_k=\boldsymbol{M}^{1/2}\boldsymbol{x}_k,$$
$$\hat{\boldsymbol{r}}_k=\boldsymbol{M}^{-1/2}\boldsymbol{r}_k,$$
$$\hat{\boldsymbol{p}}_k=\boldsymbol{M}^{1/2}\boldsymbol{p}_k.$$

所以，$\boldsymbol{x}_k$收敛于$\boldsymbol{M}^{-1/2}$乘$\hat{\boldsymbol{A}}\hat{\boldsymbol{x}}=\hat{\boldsymbol{b}}$的解，即$\boldsymbol{M}^{-1/2}\hat{\boldsymbol{A}}^{-1}\hat{\boldsymbol{b}}=\boldsymbol{A}^{-1}\boldsymbol{b}$.

证明见问题6.14.

现在描述一些通常的预条件子. 注意极小化 $M^{-1}A$ 的条件数和保持 $Mx=b$ 容易求解这一对目标是相互冲突的：选择 $M=A$ 极小化 $M^{-1}A$ 的条件数但 $Mx=b$ 维持像原问题一样难于求解. 选择 $M=I$ 使求解 $Mx=b$ 是平凡的但 $M^{-1}A$ 的条件数维持不变. 因为在算法的内循环中需要求解 $Mx=b$，所以限定讨论 $Mx=b$ 容易求解的那些 M，并描述什么时候它们可能减小 $M^{-1}A$ 的条件数.

- 若 A 有变化相差很大的对角元，则可使用简单的对角预条件子 $M=\mathrm{diag}(a_{11},\cdots,a_{nn})$. 可以证明在所有可能的对角预条件子中，这种选择减少 $M^{-1}A$ 的条件数不超出其最小值的 n 倍[244]. 这也称为雅可比预条件.
- 作为第一种预条件子的推广，设

$$A=\begin{bmatrix} A_{11} & \cdots & A_{1k} \\ \vdots & & \vdots \\ A_{k1} & \cdots & A_{kk} \end{bmatrix}$$

 是一个分块矩阵，其中对角块 A_{ii} 是方阵. 则在所有块对角预条件子

$$M=\begin{bmatrix} M_{11} & & \\ & \ddots & \\ & & M_{kk} \end{bmatrix}$$

 中，其中 M_{ii} 和 A_{ii} 有相同的维数，选择 $M_{ii}=A_{ii}$ 极小化 $M^{-1/2}AM^{-1/2}$ 的条件数不超过 k 倍[69]. 这也称为块雅可比预条件.
- 类似于雅可比，也可用 SSOR 产生一个(块)预条件子.
- A 的一个不完全楚列斯基分解 LL^{T} 是 $A\approx LL^{\mathrm{T}}$ 的一个近似，其中 L 限定如 A 原来的模式那样一个特殊的稀疏模式. 换言之，在楚列斯基分解中不允许填补，然后使用 $M=LL^{\mathrm{T}}$.（对非对称问题，有一个相应的不完全 LU 预条件子）. [318]
- 当 A 代表在一个自然区域 Ω 上的方程(例如泊松方程)时，使用区域分解. 迄今为止，对泊松方程设 Ω 是单位正方形. 更一般地，可以把区域 Ω 分裂成不相交的(或稍稍交叠的)子区域 $\Omega=\cup_j\Omega_j$，而且可以在每个子区域上独立地求解方程. 例如，若我们正在求解泊松方程而且若子区域是方的或长方的，则子问题利用 FFT 可以很快地求解(见 6.7 节). 求解这些子问题对应于一个块对角阵 M（若子区域不相交)或块对角阵 M 之积(若子区域交叠). 这在 6.10 节中更详尽地讨论.

在软件包 PETSc[232]和 PARPRE(NETLIB/scalapack/parpre. tar. gz)中提供了许多这类预条件子.

6.6.6 解 $Ax=b$ 的其他克雷洛夫子空间算法

迄今为止集中讨论了对称正定线性方程组和极小化残差的 A^{-1}－范数. 本节中描

述其他类型线性方程组的方法并基于矩阵的简单性质对使用哪个方法提供建议. 概要见图6-8[15,107,136,214], 而有关的细节, 特别是关于选择方法和软件更全面的建议见 NETLIB/templates.

通过求解正规方程 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$(或 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}=\boldsymbol{b},\boldsymbol{x}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$) 可以把任何方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 改变成对称正定方程组. 这包括最小二乘问题 $\min_x\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2$. 倘若可同时用 $\boldsymbol{A}$ 和 $\boldsymbol{A}^{\mathrm{T}}$ 乘向量, 则就允许我们使用 CG. 因为 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 和 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 的条件数是 $\boldsymbol{A}$ 的条件数的平方, 所以当 $\boldsymbol{A}$ 病态时这个方法可能导致缓慢的收敛, 但当 $\boldsymbol{A}$ 良态(或者如6.6.4节中讨论的那样 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 有"好的"特征值分布)时这个方法是快速收敛的.

当 $\boldsymbol{A}$ 对称正定时, 我们可以极小化残差的2-范数而不是 $\boldsymbol{A}^{-1}$-范数. 这称为极小残差算法, 或 MINRES[194]. 因为 MINRES 比 CG 更昂贵, 所以它不用于正定方程组. 但是当矩阵对称不定时, 不能使用 CG 然而可以使用 MINRES. 此时, 也可使用 Paige 和 Saunders 的 SYMMLQ 算法[194], 在每一步, 算法产生一个残差, $r_k\perp K_k(\boldsymbol{A},\boldsymbol{b})$

319　遗憾的是, 除对称阵外只有极少数矩阵存在类似于 CG 的算法它们同时满足

1. 或者极小化残差 $\|r_k\|_2$ 或者保持它正交 $r_k\perp K_k$.
2. 需要与 k 无关的固定数量的点积和内循环中的 saxpy.

满足这两个性质的算法基本上只对形如 $\mathrm{e}^{i\theta}(T+\sigma I)$ 的矩阵存在, 其中 $T=T^{\mathrm{T}}$(或 $TH=(HT)^{\mathrm{T}}$, H 为某个对称正定阵), θ 是实的而 σ 是复的[102,251]. 对这些对称和特殊的非对称阵 $\boldsymbol{A}$, 它证实为计算 $K_k(A,b)$ 的正交基 $[q_1,\cdots,q_k]$ 我们可找到如同兰乔斯算法中一样的短递归. 在短递归中更新 q_k 只有少数几项的事实表明可以非常有效地计算它.

对一般的非对称阵 $\boldsymbol{A}$ 这个短递归的存在性不再成立. 此时可使用阿诺尔迪算法. 因而代替三对角阵 $\boldsymbol{T}_k=\boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_k$, 得到一个完全的上海森伯格阵 $\boldsymbol{H}_k=\boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_k$. GMRES (广义的极小残差) 算法利用这个分解去选择 $\boldsymbol{x}_k=\boldsymbol{Q}_k\boldsymbol{y}_k\in K_k(A,b)$ 使下列残差达到极小

$$
\begin{aligned}
\|\boldsymbol{r}_k\|_2 &= \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k\|_2 \\
&= \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{Q}_k\boldsymbol{y}_k\|_2 \\
&= \|\boldsymbol{b}-(\boldsymbol{Q}\boldsymbol{H}\boldsymbol{Q}^{\mathrm{T}})\,\boldsymbol{Q}_k\boldsymbol{y}_k\|_2 \qquad \text{由(6.30)式} \\
&= \|\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}-\boldsymbol{H}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}_k\boldsymbol{y}_k\|_2 \qquad \text{因为 } \boldsymbol{Q} \text{ 是正交的} \\
&= \left\| \boldsymbol{e}_1\|\boldsymbol{b}\|_2 - \begin{bmatrix} \boldsymbol{H}_k & \boldsymbol{H}_{uk} \\ \boldsymbol{H}_{ku} & \boldsymbol{H}_u \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{y}_k \\ 0 \end{bmatrix} \right\|_2 \\
&\qquad \text{由(6.30)式及因为 } \boldsymbol{Q}=[\boldsymbol{Q}_k,\boldsymbol{Q}_u] \text{ 的第一列是 } \boldsymbol{b}/\|\boldsymbol{b}\|_2 \\
&= \left\| \boldsymbol{e}_1\|\boldsymbol{b}\|_2 - \begin{bmatrix} \boldsymbol{H}_k \\ \boldsymbol{H}_{ku} \end{bmatrix} \boldsymbol{y}_k \right\|_2 .
\end{aligned}
$$

因为只有 $\boldsymbol{H}_{ku}$ 的第一列非零, 所以这是关于 $\boldsymbol{y}_k$ 的元素的 $(k+1)\times k$ 上海森伯格最

小二乘问题. 因为它是上海森伯格阵，所以求解它所需的 QR 分解可以用 k 个吉文斯旋转实现，代价为 $O(k^2)$ 而不是 $O(k^3)$. 还有所需的存储量为 $O(kn)$，因为 Q_k 必须存放. 限制花费和存储量增长的一个方法是*重新开始* GMRES，即取 k 步之后算出的解答 x_k 重新开始 GMRES 求解线性方程组 $\boldsymbol{Ad}=\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{Ax}_k$，更新解得到 $\boldsymbol{x}_k+\boldsymbol{d}$；这称为 GMRES($k$). 尽管如此，GMRES($k$)还是比 CG 更昂贵，其中内循环的花费根本不依赖于 k.

非对称线性方程组的另一种方法是放弃计算 $K_k(\boldsymbol{A},\boldsymbol{b})$ 的标准正交基而再约化 $\boldsymbol{A}$ 为(非对称)三对角形式计算非标准正交基. 这称为*非对称兰乔斯方法*，并需要用 $\boldsymbol{A}$ 和 $\boldsymbol{A}^{\mathrm{T}}$ 两个的矩阵-向量乘法. 因为 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{z}$ 有时计算较困难(或不可能)，所以这是重要的(见例 6.13). 三对角形式的优点是用三对角阵求解比用海森伯格阵求解更容易. 缺点是基向量可能非常病态并且事实上可能根本不存在. 这是一个称之为*故障*(breakdown)的现象. 潜在的效率导致大量的关于避免或缓和这个不稳定性(超前兰乔斯)和对抗方法，包括*双共轭梯度*和*拟-极小残差*的研究. 还有一些不需要用 A^{T} 相乘的形式，包括*平方共轭梯度*和*双共轭梯度稳定化*(bi-conjugate gradient stabilized). 没有一种方法在所有情况下都是最佳的.

320

图 6-8 是一棵决策树，假定我们没有矩阵 $\boldsymbol{A}$ 的其他深入的知识.(例如它从泊松方程中产生)，那么这棵决策树对首先尝试哪种方法给出一个简单的忠告.

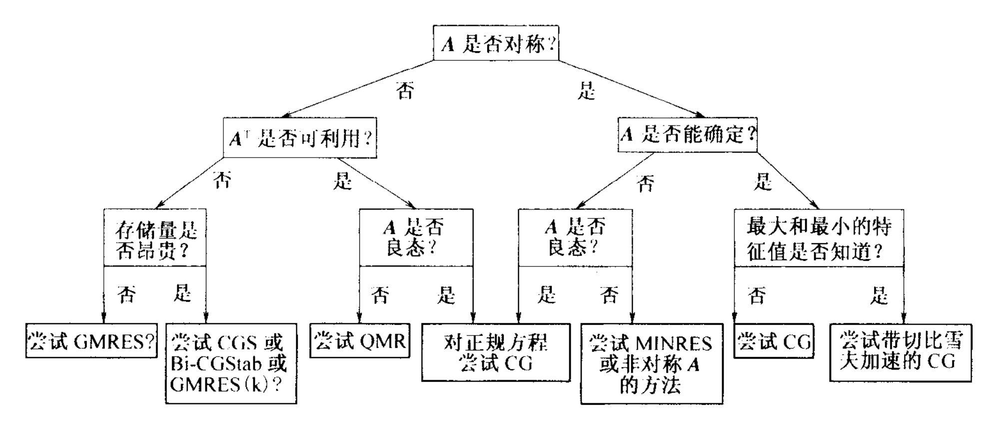


图 6-8 对 $Ax=b$ 选择一种迭代算法的决策树. Bi-CGStab = 双共轭梯度稳定化. QMR = 拟极小残差. CGS = 平方 CG.

6.7 快速傅里叶变换

本节中的 i 始终表示 $\sqrt{-1}$.

我们通过指出如何以一种需要用 $\boldsymbol{T}_N$ 的特征向量矩阵乘法运算的方法求解二维泊松方程开始讨论. 这种矩阵-矩阵乘法的直接执行过程将花费 $O(N^3)=O(n^{3/2})$ 次运

算，这个代价是昂贵的. 然后指出如何利用 FFT 执行这个乘法只要 $O(N^2\log N)=O(n\log n)$次运算，这个代价不超过最佳的 $\log n$ 倍.

这个解是原来的微分方程(6.1)或(6.6)的傅里叶级数解的离散模拟. 后面我们将更精确地获得这个模拟.

321

设 $T_N=Z\Lambda Z^{\mathrm{T}}$ 是如引理6.1中定义的 T_N 的特征分解. 首先用(6.11)式中的二维泊松方程的公式:

$$T_NV+VT_N=h^2F.$$

代入 $T_N=Z\Lambda Z^{\mathrm{T}}$ 以及用 Z^{T} 左乘和用 Z 右乘得到

$$Z^{\mathrm{T}}(Z\Lambda Z^{\mathrm{T}})VZ+Z^{\mathrm{T}}V(Z\Lambda Z^{\mathrm{T}})Z=Z^{\mathrm{T}}(h^2F)Z$$

或

$$\Lambda V'+V'\Lambda=h^2F',$$

其中 $V'=Z^{\mathrm{T}}VZ$ 和 $F'=Z^{\mathrm{T}}FZ$. 这个最后等式的(j,k)元素是

$$(\Lambda V'+V'\Lambda)_{jk}=\lambda_j v'_{jk}+v'_{jk}\lambda_k=h^2f'_{jk},$$

由此可解出 v'_{jk}得到

$$v'_{jk}=\frac{h^2f'_{jk}}{\lambda_j+\lambda_k}.$$

这就得到算法的第一种形式.

算法 6.13 利用特征分解 $T_N=Z\Lambda Z^{\mathrm{T}}$ 求解二维泊松方程:

1) $F'=Z^{\mathrm{T}}FZ$

2) 对一切 j 和 k, $v'_{jk}=\dfrac{h^2f'_{jk}}{\lambda_j+\lambda_k}$

3) $V=ZV'Z^{\mathrm{T}}$

第2步的花费是 $3N^2=3n$ 次运算，而第1步和第3步是用 Z 及 $Z^{\mathrm{T}}=Z$ 的4次矩阵-矩阵乘法，利用常规的算法总共 $8N^3=8n^{3/2}$次运算. 在下节中指出用 Z 相乘本质上怎么与计算离散傅里叶变换(discrete Fourier transform)相同，利用 FFT 它可以用 $O(N^2\log N)=O(n\log n)$次运算完成.

(利用6.3.3节中引入的克罗内克积的语言，以及特别来自命题6.1的 $T_{N\times N}$的特征分解,

$$T_{N\times N}=I\otimes T_N+T_N\otimes I=(Z\otimes Z)\cdot(I\otimes\Lambda+\Lambda\otimes I)\cdot(Z\otimes Z)^{\mathrm{T}},$$

我们可如下改写公式说明算法6.13是正确的:

$$\begin{aligned}\mathrm{vec}(V)&=(T_{N\times N})^{-1}\cdot\mathrm{vec}(h^2F)\\&=((Z\otimes Z)\cdot(I\otimes\Lambda+\Lambda\otimes I)\cdot(Z\otimes Z)^{\mathrm{T}})^{-1}\cdot\mathrm{vec}(h^2F)\\&=(Z\otimes Z)^{-\mathrm{T}}\cdot(I\otimes\Lambda+\Lambda\otimes I)^{-1}\cdot(Z\otimes Z)^{-1}\cdot\mathrm{vec}(h^2F)\\&=(Z\otimes Z)\cdot(I\otimes\Lambda+\Lambda\otimes I)^{-1}\cdot(Z^{\mathrm{T}}\otimes Z^{\mathrm{T}})\cdot\mathrm{vec}(h^2F).\end{aligned}\tag{6.47}$$

我们断言从右到左做所指出的矩阵-向量乘法在数学上与算法6.13是相同的；见问题6.9. 这也表明如何把算法推广到高维的泊松方程.）

322

6.7.1 离散傅里叶变换

在本小节中，将把矩阵的行和列从0到$N-1$而不是从1到N编号.

定义6.17 N维向量$\boldsymbol{x}$的离散傅里叶变换(DFT)是向量$\boldsymbol{y}=\boldsymbol{\Phi x}$，其中$\boldsymbol{\Phi}$是如下定义的$N\times N$阶矩阵. 设$\omega=e^{\frac{-2\pi i}{N}}=\cos\frac{2\pi}{N}-i\cdot\sin\frac{2\pi}{N}$是主$N$次单位根. 则$\boldsymbol{\Phi}_{jk}=\omega^{jk}$. $\boldsymbol{y}$的反离散傅里叶变换(IDFT)是向量$\boldsymbol{x}=\boldsymbol{\Phi}^{-1}\boldsymbol{y}$.

引理6.9 $\frac{1}{\sqrt{N}}\boldsymbol{\Phi}$是对称酉阵，故$\boldsymbol{\Phi}^{-1}=\frac{1}{N}\boldsymbol{\Phi}^*=\frac{1}{N}\overline{\boldsymbol{\Phi}}$.

证明 显然，$\boldsymbol{\Phi}=\boldsymbol{\Phi}^{T}$，故$\overline{\boldsymbol{\Phi}}=\boldsymbol{\Phi}^*$，我们只需证明$\boldsymbol{\Phi}\cdot\overline{\boldsymbol{\Phi}}=N\cdot\boldsymbol{I}$. 计算$(\boldsymbol{\Phi}\overline{\boldsymbol{\Phi}})_{lj}=\sum_{k=0}^{N-1}\phi_{lk}\overline{\phi}_{kj}=\sum_{k=0}^{N-1}\omega^{lk}\overline{\omega}^{kj}=\sum_{k=0}^{N-1}\omega^{k(l-j)}$，因为$\overline{\omega}=\omega^{-1}$. 若$l=j$，则这个和显然为$N$. 若$l\neq j$，这是一个几何级数求和，其值为$\frac{1-\omega^{N(l-j)}}{1-\omega^{l-j}}=0$，因为$\omega^N=1$. □

这样，DFT和IDFT两者刚好都是矩阵-向量乘法，并且可用$2N^2$flops直接执行. 这个运算称为DFT是因为它与其他两类傅里叶分析有密切的数学关系：

傅里叶变换及其逆	$F(\zeta)=\int_{-\infty}^{\infty}e^{-2\pi i\zeta x}f(x)dx, f(x)=\int_{-\infty}^{\infty}e^{+2\pi i\zeta x}F(\zeta)d\zeta$
傅里叶级数及其逆，其中f在[0,1]上是周期的	$c_j=\int_0^1 e^{-2\pi ijx}f(x)dx, f(x)=\sum_{j=-\infty}^{\infty}e^{+2\pi ijx}c_j$
DFT及其逆	$y_j=(\Phi x)_j=\sum_{k=0}^{N-1}e^{-2\pi ijk/N}x_k, x_k=(\Phi^{-1}y)_k=\frac{1}{N}\sum_{j=0}^{N-1}e^{+2\pi ijk/N}y_j$

我们将以两个途径获得这个密切的更具体的关系. 第一，将指出如何利用DFT求解模型问题然后利用傅里叶级数求解原来的泊松方程(6.1). 这个例子将促使我们寻找一个用Φ相乘的快速方法. 这个快速方法称为快速傅里叶变换(fast Fourier transform)或FFT. 它只需要$\frac{3}{2}N\log_2N$flops左右而不是$2N^2$flops，其次，通过强调不同类型傅里叶分析之间共享的另一个数学关系：把卷积化为乘法导出FFT.

在算法6.13中指出求解离散的泊松方程$\boldsymbol{T}_N\boldsymbol{V}+\boldsymbol{V}\boldsymbol{T}_N=h^2\boldsymbol{F}$的$\boldsymbol{V}$需要用$N\times N$阶矩阵$\boldsymbol{Z}$相乘的能力，其中

$$z_{jk}=\sqrt{\frac{2}{N+1}}\sin\frac{\pi(j+1)(k+1)}{N+1}.$$

323

（记住在本节中行和列从0到$N-1$编号）. 现在考虑$(2N+2)\times(2N+2)$阶DFT矩阵$\boldsymbol{\Phi}$，它的j，k元素为

$$\exp\left(\frac{-2\pi ijk}{2N+2}\right)=\exp\left(\frac{-\pi ijk}{N+1}\right)=\cos\frac{\pi jk}{N+1}-i\cdot\sin\frac{\pi jk}{N+1}.$$

因而 $N\times N$ 阶矩阵 $\mathbf{Z}$，由 $-\sqrt{\frac{2}{N+1}}$ 乘以 $\mathbf{\Phi}$ 的第二到第$(N+1)$行和列的虚部组成．如果能利用 FFT 通过 $\mathbf{\Phi}$ 有效地相乘，则我们可以通过 $\mathbf{Z}$ 有效地相乘．（为更加有效起见，修改 FFT 算法，直接用 $\mathbf{Z}$ 相乘，这个算法在下面描述称为快速正弦变换(fast sine transform)．但是也可以只使用 FFT）．因而，快速相乘 $\boldsymbol{ZF}$ 需要对 $\boldsymbol{F}$ 的每列作一个 FFT-类运算，而快速相乘 $\boldsymbol{FZ}$ 需要对 F 的每行作相同的运算．（在三维情况，将设 V 是未知量的 $N\times N\times N$ 数组，对每一个平行于坐标轴的 $3N^2$ 截面实施相同的运算）．

6.7.2　用傅里叶级数解连续模型问题

现在返回到矩阵的行和列从 1 到 N 编号．

在本节中将指出求解离散的模型问题的算法是利用傅里叶级数求解原来的微分方程(6.1)的一个自然的模拟．我们将对一维模型问题讨论这个结论．

记得在[0,1]上的泊松方程是 $-\frac{\mathrm{d}^2v}{\mathrm{d}x^2}=f(x)$ 带边界条件 $v(0)=v(1)$．为求解这个方程，将傅里叶级数展开 $v(x)$：$v(x)=\sum_{j=1}^{\infty}\alpha_j\sin(j\pi x)$（边界条件 $v(1)=0$ 告诉我们不出现余弦项）．把 $v(x)$ 与泊松方程连结起来得到

$$\sum_{j=1}^{\infty}\alpha_j(j^2\pi^2)\sin(j\pi x)=f(x).$$

用 $\sin(k\pi x)$ 乘两边，从 0 到 1 积分并利用 $\int_0^1\sin(j\pi x)\sin(k\pi x)\,\mathrm{d}x=0, j\neq k$，以及 $\int_0^1\sin(j\pi x)\sin(k\pi x)\,\mathrm{d}x=1/2$，$j=k$，得到

$$\alpha_k=\frac{2}{k^2\pi^2}\int_0^1\sin(k\pi x)f(x)\,\mathrm{d}x$$

并且最后得到

$$v(x)=\sum_{j=1}^{\infty}\left(\frac{2}{j^2\pi^2}\int_0^1\sin(j\pi y)f(y)\,\mathrm{d}y\right)\sin(j\pi x).\tag{6.48}$$

现在考虑离散的模型问题 $\boldsymbol{T}_Nv=h^2f$．因为 $\boldsymbol{T}_N=\mathbf{Z\Lambda Z}^{\mathrm{T}}$，我们可以记 $v=\boldsymbol{T}_N^{-1}h^2f=\mathbf{Z\Lambda}^{-1}\mathbf{Z}^{\mathrm{T}}h^2f$，故

324

$$v_k=\sum_{j=1}^{N}z_{kj}\frac{h^2}{\lambda_j}(\mathbf{Z}^{\mathrm{T}}f)_j=\sum_{j=1}^{N}\sin\frac{\pi jk}{N+1}\left(\frac{h^2}{\lambda_j}\sqrt{\frac{2}{N+1}}(\mathbf{Z}^{\mathrm{T}}f)_j\right),\tag{6.49}$$

其中

$$\begin{aligned}\sqrt{\frac{2}{N+1}}(\mathbf{Z}^{\mathrm{T}}f)_j&=\sqrt{\frac{2}{N+1}}\sum_{l=1}^{N}\sqrt{\frac{2}{N+1}}\sin\left(\frac{\pi jl}{N+1}\right)f_l\\&=2\sum_{l=1}^{N}\frac{1}{N+1}\sin\left(\frac{\pi jl}{N+1}\right)f_l\\&\approx2\int_0^1\sin(\pi jy)f(y)\,\mathrm{d}y,\end{aligned}$$

因为最后的和式正好是积分的黎曼和近似. 而且对小的 j, 记住 $\frac{h^2}{\lambda_j} \approx \frac{1}{j^2 \pi^2}$. 因而我们看出离散问题(6.49)的解如何逼近连续问题(6.48)的解, 用 Z^T 相乘对应于用 $\sin(j\pi x)$ 相乘并积分, 而用 Z 相乘对应于不同的傅里叶分量求和.

6.7.3 卷积

卷积是傅里叶分析中的一个重要的运算, 其定义依赖于我们正在做的是傅里叶变换, 傅里叶级数还是 DFT:

傅里叶变换	$(f*g)(x) \equiv \int_{-\infty}^{\infty} f(x-y)g(y)\mathrm{d}y$
傅里叶级数	$(f*g)(x) \equiv \int_0^1 f(x-y)g(y)\mathrm{d}y$
DFT	若 $a=[a_0,\cdots,a_{N-1},0,\cdots,0]^T$ 和 $b=[b_0,\cdots,b_{N-1},0,\cdots,0]^T$ 是 $2N$ 维向量, 则 $a*b \equiv c=[c_0,\cdots,c_{2N-1}]^T$,其中 $c_k=\sum_{j=0}^{k} a_j b_{k-j}$

为了说明离散卷积的用途, 考虑多项式乘法. 设 $\mathrm{a}(x)=\sum_{k=0}^{N-1} a_k x^k$ 和 $\mathrm{b}(x)=\sum_{k=0}^{N-1} b_k x^k$ 是$(N-1)$次多项式. 则它们的积 $c(x) \equiv \mathrm{a}(x)\cdot\mathrm{b}(x)=\sum_{k=0}^{2N-1} c_k x^k$, 其中系数 $c_0,\cdots,c_{2N-1}$ 由离散卷积给出.

傅里叶变换, 傅里叶级数或 DFT 的一个目的是把卷积转换成乘法. 在傅里叶变换情况, $\mathcal{F}(f*g)=\mathcal{F}(f)\cdot\mathcal{F}(g)$; 即卷积的傅里叶变换是傅里叶变换之积. 在傅里叶级数情况, $c_j(f*g)=c_j(f)\cdot c_j(g)$; 即卷积的傅里叶系数是傅里叶系数之积. 对离散卷积这个结论同样成立.

定理 6.10 设 $a=[a_0,\cdots,a_{N-1},0,\cdots,0]^T$ 和 $b=[b_0,\cdots,b_{N-1},0,\cdots,0]^T$ 是 $2N$ 维向量, 并设 $c=a*b=[c_0,\cdots,c_{2N-1}]^T$. 则 $(\Phi c)_k=(\Phi a)_k\cdot(\Phi b)_k$.

325

证明 若 $a'=\Phi a$, 则多项式 $\mathrm{a}(x)=\sum_{j=0}^{N-1} a_j x^j$ 在 $x=\omega^k$ 上的值 $a'_k=\sum_{j=0}^{2N-1} a_j \omega^{kj}$. 类似地, $b'=\Phi b$ 意味着 $b'_k=\sum_{j=0}^{N-1} b_j \omega^{kj}=\mathrm{b}(\omega^k)$ 和 $c'=\Phi c$ 意味着 $c'_k=\sum_{j=0}^{2N-1} c_j \omega^{kj}=c(\omega^k)$. 所以

$$a'_k\cdot b'_k=\mathrm{a}(\omega^k)\cdot\mathrm{b}(\omega^k)=\mathrm{c}(\omega^k)=c'_k$$

证毕. □

换言之, DFT 是多项式在点 $\omega^0,\cdots,\omega^{N-1}$ 上求值, 相反地 IDFT 是多项式插值, 产生一个多项式的系数, 给出它在 $\omega^0,\cdots,\omega^{N-1}$ 上的值.

6.7.4 计算快速傅里叶变换

我们将根据刚才讨论的把 FFT 解释为多项式求值来导出 FFT. 目标是对 $0\le j\le N-1$, 在 $x=\omega^j$ 上计算 $\mathrm{a}(x)=\sum_{k=0}^{N-1} a_k x^k$. 为简单起见, 假设 $N=2^m$. 现

在记

$$\begin{aligned} \mathrm{a}(x) &= a_0 + a_1 x + a_2 x^2 + \cdots + a_{N-1} x^{N-1} \\ &= (a_0 + a_2 x^2 + a_4 x^4 + \cdots) + x(a_1 + a_3 x^2 + a_5 x^4 + \cdots) \\ &\equiv \mathrm{a}_{偶}(x^2) + x \cdot \mathrm{a}_{奇}(x^2). \end{aligned}$$

因而，需要对 $0 \leqslant j \leqslant N-1$，在 $(\omega^j)^2$ 上计算两个次数为 $\frac{N}{2}-1$ 的多项式 $\mathrm{a}_{偶}$ 和 $\mathrm{a}_{奇}$. 但这实际上正好是 $\frac{N}{2}$ 个点 ω^{2j}，$0 \leqslant j \leqslant \frac{N}{2}-1$，因为 $\omega^{2j} = \omega^{2\left(j+\frac{N}{2}\right)}$.

因而在所有 N 个 N 次单位根上计算一个 $N-1=2^m-1$ 次多项式与在所有 $\frac{N}{2}$ 个 $\frac{N}{2}$ 次单位根上计算两个 $\frac{N}{2}-1$ 次多项式，然后用 N 次乘法和加法组合这些结果是相同的. 这个计算可以递归地完成.

算法 6.14　FFT(递归形式)：

```
function FFT(a,N)
    if N = 1
        return a
    else
        a'偶 = FFT(a偶, N/2)
        a'奇 = FFT(a奇, N/2)
        ω = e^(-2πi/N)
        ω = [ω^0, ···, ω^(N/2-1)]
        return a' = [a'偶 + ω. * a'奇, a'偶 - ω. * a'奇]
    endif
```

这里. * 意味着数组的按分量乘法(如 Matlab 中一样)，并且使用了 $\omega^{j+N/2} = -\omega^j$ 的事实.

326

设这个算法的代价用 $C(N)$ 表示. 则我们看出 $C(N)$ 满足递归式 $C(N)=2C(N/2)+3N/2$(假定 ω 的乘幂预先算出并存储在表中). 为求解这个递归式，记

$$\begin{aligned} C(N) &= 2C\left(\frac{N}{2}\right) + \frac{3N}{2} = 4C\left(\frac{N}{4}\right) + 2 \cdot \frac{3N}{2} = 8C\left(\frac{N}{8}\right) + 3 \cdot \frac{3N}{2} \\ &= \cdots \\ &= \log_2 N \cdot \frac{3N}{2}. \end{aligned}$$

所以计算 $N \times N$ 阶矩阵的每列(或每行)的 FFT 花费 $\log_2 N \cdot \frac{3N^2}{2}$. 这个复杂性分

析证实表6-1中关于FFT的表列数据是正确的.

实际上，为了尽可能有效，FFT的执行过程利用简单的嵌套循环而不是递归；见NETLIB/fftpack. 另外，这个执行过程有时以位-相反(bit-reversed)次序获得分量. 这意味着代替获得 $y_0, y_1, \cdots, y_{N-1}$，其中 $y=\Phi x$，对下标 j 重新排序使位模式相反. 例如，若 $N=8$，下标从 $0=000_2$ 运行到 $7=111_2$. 下表指明正规次序和位-相反次序：

正规递增次序	位-相反次序
$0=000_2$	$0=000_2$
$1=001_2$	$4=100_2$
$2=010_2$	$2=010_2$
$3=011_2$	$6=110_2$
$4=100_2$	$1=001_2$
$5=101_2$	$5=101_2$
$6=110_2$	$3=011_2$
$7=111_2$	$7=111_2$

反FFT取消这个重新排序并以它们原来的次序获得结果. 倘若通过适当的划分特征值使其下标对应于位-相反次序，那么这些算法可用于求解模型问题(注意Matlab总以正规递增次序获得结果).

6.8 块循环约化

块循环约化是模型问题的另一个快速($O(N^2\log_2 N)$)方法，然而它比基于FFT的解应用稍为更广泛一点. 在向量计算机上关于模型问题的最快的算法通常是块循环约化和FFT的混合体.

首先描述一个简单的但数值不稳定的算法形式，然后讲一点如何稳定它有关情况. 记模型问题为

327

$$\begin{bmatrix} A & -I & & \\ -I & \ddots & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & A \end{bmatrix}\begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}=\begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix},$$

其中 $A=T_N+2I_N$ 的维数 N 假定为奇数. 注意 x_i 和 b_i 也是 N 维向量.

利用块高斯消元法组合三个联贯的方程的集合

$$\begin{array}{rlcccccl} + & [& -x_{j-2} & +Ax_{j-1} & -x_j & & = b_{j-1} &], \\ +A* & [& & -x_{j-1} & +Ax_j & -x_{j+1} & = b_j &], \\ + & [& & & -x_j & +Ax_{j+1} \quad -x_{j+2} & = b_{j+1} &], \end{array}$$

于是消去 x_{j-1} 和 x_{j+1}：

$$-x_{j-2}+(A^2-2I)x_j-x_{j+2}=b_{j-1}+Ab_j+b_{j+1}.$$

对每三个联贯的方程的集合做这个消元法得到两个方程组：一个是关于 j 为偶数的 x_j，

$$\begin{bmatrix} B & -I & & & \\ -I & B & -I & & \\ & -I & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & B \end{bmatrix}\begin{bmatrix} x_2 \\ x_4 \\ \vdots \\ x_{N-1} \end{bmatrix}=\begin{bmatrix} b_1+Ab_2+b_3 \\ b_3+Ab_4+b_5 \\ \vdots \\ b_{N-2}+Ab_{N-1}+b_N \end{bmatrix}, \tag{6.50}$$

其中 $B=A^2-2I$，而一组方程是关于 j 为奇数的 x_j，由此在求解方程(6.50)之后，我们可以求解 j 为奇数的 x_j：

$$\begin{bmatrix} A & & & \\ & A & & \\ & & \ddots & \\ & & & A \end{bmatrix}\begin{bmatrix} x_1 \\ x_3 \\ \vdots \\ x_N \end{bmatrix}=\begin{bmatrix} b_1+x_2 \\ b_3+x_2+x_4 \\ \vdots \\ b_N+x_{N-1} \end{bmatrix}.$$

注意方程(6.50)具有与原来的问题相同的形式，故可递归地重复这个过程. 例如，在下一步我们得到

$$\begin{bmatrix} C & -I & & & \\ -I & C & -I & & \\ & -I & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & C \end{bmatrix}\begin{bmatrix} x_4 \\ x_8 \\ \vdots \end{bmatrix}=\begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix},\text{其中 } C=B^2-2I,$$

并且

$$\begin{bmatrix} B & & & \\ & B & & \\ & & \ddots & \\ & & & B \end{bmatrix}\begin{bmatrix} x_2 \\ x_6 \\ \vdots \end{bmatrix}=\begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}.$$

重复这个过程直到留下一个方程，这个方程用另外的方法求解.

正式形成这个算法如下：假定 $N=N_0=2^{k+1}-1$，且设 $N_r=2^{k+1-r}-1$. 设 $A^{(0)}=A$ 和 $b_j^{(0)}=b_j, j=1,\cdots,N$.

算法 6.15 块循环约化：

1) 约化：

for $r=0$ to $k-1$

$A^{(r+1)}=(A^{(r)})^2-2I$

for $j=1$ to N_{r+1}

（续）

$$b_j^{(r+1)} = b_{2j-1}^{(r)} + \boldsymbol{A}^{(r)} b_{2j}^{(r)} + b_{2j+1}^{(r)}$$

end for

end for

注释：在第 r 步问题化为

$$\begin{bmatrix} \boldsymbol{A}^{(r)} & -\boldsymbol{I} & & \\ -\boldsymbol{I} & \ddots & \ddots & \\ & \ddots & \ddots & -\boldsymbol{I} \\ & & -\boldsymbol{I} & \boldsymbol{A}^{(r)} \end{bmatrix} \begin{bmatrix} x_1^{(r)} \\ \vdots \\ x_{N_r}^{(r)} \end{bmatrix} = \begin{bmatrix} b_1^{(r)} \\ \vdots \\ b_{N_r}^{(r)} \end{bmatrix}$$

2）用另外的方法求解 $\boldsymbol{A}^{(k)} \boldsymbol{x}^{(k)} = \boldsymbol{b}^{(k)}$.

3）向后求解：

for $r = k-1, \cdots, 0$

for $j = 1$ to N_{r+1}

$x_{2j}^{(r)} = x_j^{(r+1)}$

end for

for $j = 1$ to N_r 第 2 步

求解 $A^{(r)} x_j^{(r)} = b_j^{(r)} + x_{j-1}^{(r)} + x_{j+1}^{(r)}$ 得 $x_j^{(r)}$

（取 $x_0^{(r)} = x_{N_{r=1}}^{(r)} \equiv 0$）

end for

end for

最后，$x = x^{(0)}$ 是要求的结果.

329

这个简单的方法有两个缺点：

1）因为 $\boldsymbol{A}^{(r)}$ 快速增长：$\|\boldsymbol{A}^{(r)}\| \sim \|\boldsymbol{A}^{(r-1)}\|^2 \approx 4^{2^r}$，所以它是数值不稳定的，故在计算 $b_j^{(r+1)}$ 中，$b_{2j\pm1}^{(r)}$ 在舍入中被丢失.

2）若 $\boldsymbol{A}$ 是三对角阵，则 $\boldsymbol{A}^{(r)}$ 的带宽为 2^r+1，故它马上变成稠密，因而乘法和求解代价昂贵.

下面是第二个缺点的一个修正. 注意 $\boldsymbol{A}^{(r)}$ 是一个 2^r 次的多项式 $p_r(\boldsymbol{A})$：

$$p_0(\boldsymbol{A}) = \boldsymbol{A} \text{ 而 } p_{r+1}(\boldsymbol{A}) = (p_r(\boldsymbol{A}))^2 - 2\boldsymbol{I}.$$

引理 6.10 设 $t = 2\cos\theta$. 则 $p_r(t) = p_r(2\cos\theta) = 2\cos(2^r\theta)$.

证明 这是一个简单的三角恒等式. □

注意 $p_r(t) = 2\cos\left(2^r \arccos\left(\frac{t}{2}\right)\right) = 2T_{2^r}\left(\frac{t}{2}\right)$，其中 $T_{2^r}(\cdot)$ 是切比雪夫多项式（见 6.5.6 节）.

引理6.11　$p_r(t)=\prod_{j=1}^{2^r}(t-t_j)$，其中 $t_j=2\cos\left(\pi\frac{2j-1}{2^r}\right)$.

证明　切比雪夫多项式的零点在引理6.7中给出.　□

因而 $A^{(r)}=\prod_{j=1}^{2^r}\left(A-2\cos\left(\pi\frac{2j-1}{2^r}\right)\right)$，故求解 $A^{(r)}z=c$ 等价于求解具有三对角系数阵 $A+2\cos\left(\pi\frac{2j-1}{2^r}\right)$ 的 2^r 个三对角方程组，用三对角高斯消元法或楚列斯基分解，每个三对角方程组花费 $O(N)$ flops.

为了数值稳定的算法需要更多的变化. 最后的算法是由Buneman给出的并在[47,46]中描述.

下面分析简单算法的代价；稳定的算法是类似的. 用大小为 N 的三对角阵相乘或求解大小为 N 的三对角方程组花费 $O(N)$ flops. 因为 $A^{(r)}$ 是 2^r 个三对角矩阵之积，所以用 $A^{(r)}$ 相乘或求解关于 $A^{(r)}$ 的方程组花费 $O(2^rN)$ flops. 算法第1)步的内循环花费 $\frac{N}{2^{r+1}}\cdot O(2^rN)=O(N^2)$ flops 去更新 $N_{r+1}\approx\frac{N}{2^{r+1}}$ 个向量 $b_j^{(r+1)}$. $A^{(r+1)}$ 不是明显地计算的. 因为第1)步中的循环执行 $k\approx\log_2N$ 次，所以第1)步的总的花费为 $O(N^2\log_2N)$. 因为类似的理由，第2)步花费 $O(2^kN)=O(N^2)$ flops，而第3)步花费 $O(N^2\log_2N)$ flops，总共花费 $O(N^2\log_2N)$ flops. 这就证实表6-1中块循环约化的表列数据是正确的.

这个算法推广到任意的块三对角阵，其中沿对角线具有重复的对称阵 A，沿非对线具有重复的与 A 可交换的对称阵 $F(FA=AF)$. 见问题6.10. 当求解离散像泊松方程一样的微分方程产生的线性方程组时这是一个普通的情况.

330

6.9　多重网格法

多重网格法是为像泊松方程那样的偏微分方程发明的，但它们也适用于更广泛的一类问题. 与迄今为止我们已讨论的其他迭代法相反，多重网格的收敛率独立于问题大小 N，而不是对大的问题慢下来. 作为推论，它可以 $O(n)$ 时间或每个未知量以一个常数工作量求解 n 个未知量的问题. 按隐含在 $O(\cdot)$ 里面的(适中的)常数为模，这是最佳的结果.

下面是为什么我们已讨论的其他迭代法对模型问题不能够是最佳的原因. 事实上，这对任何利用 x_m 和来自相邻的网格点的右端 b 的平均值计算近似 x_{m+1} 的任何迭代算法是成立的. 这包括雅可比，高斯-塞德尔，SOR(ω)，带切比雪夫加速的SSOR(最后三个方法用红－黑次序)，以及任何基于用矩阵 $T_{N\times N}$ 作矩阵－向量乘法的克雷洛夫子空间方法；这是因为用 $T_{N\times N}$ 乘一个向量也等价于求相邻网格点的平均值. 正如图6-9左上方指出的那样. 假如，在 31×31 网格上用单个非零元的右端 b 开始. 在相同的图右上方指出真解 x；注意它到处非零并且当远离中心时变得较小. 图6-9

的左下方指出用全零的初始解开始雅可比方法 5 步之后的解 $x_{J,5}$. 注意远离中心多于 5 个网格点上的解 $x_{J,5}$为零，因为作相邻的网格点平均，每次迭代只有一个网格点能够“传播信息”，而仅有的非零值最初是在网格的中心. 更一般地，k 步迭代之后，只有中心区的不超过 k 个网格点可能非零. 右下方图指出用“最近相邻值”方法 5 步之后能得到的最佳可能的解 $x_{\mathrm{Best},5}$：它与中心的不超过 5 个网格点上的 x 相符合并且必然地远离 0. 图形上看出误差 $x_{\mathrm{Best},5}-x$ 等于远离中心的第 6 个网格点上的 x 的大小. 这仍然是一个大的误差；通过正式的论证，可以证明，不论是否使用“最近相邻值”算法，在 $n\times n$ 网格上为误差减少小于 1 的常数倍，至少要作 $O(\log n)$步. 若要做得比 $O(\log n)$步(以及 $O(n\log n)$代价)更好，每次迭代必需多于一个网格点“传播信息”. 多重网格法通过交换较粗网格上最近相邻值执行这项工作，其中在一个粗网格上的一个最近相邻值可能比细网格上的一个最近相邻值更远.

多重网格法在两种相关的意义下利用粗网格执行*分而治之*(divide-and-conquer). 第一种，它利用取自 $N\times N$ 网格间隔不同的网格点的$(N/2)\times(N/2)$网格作为近似得到 $N\times N$ 网格的初始解. 较粗的$(N/2)\times(N/2)$网格依次用$(N/4)\times(N/4)$网格近似，等等递归地进行下去. 第二种方式多重网格是在*频域*中使用分而治之. 这需要我们把误差看作特征向量或不同频率的正弦曲线之和. 于是直觉上，我们对特殊的网格所做的工作将减小频率分量的一半误差而在较粗网格上不减小. 特别地，在特殊的网格上执行的工作——一种雅可比法的变形，把每个网格点上的解同它的相邻点的值作平均——使解更光滑，这样做等价于除去高频误差，在下页将进一步说明这些概念.

331

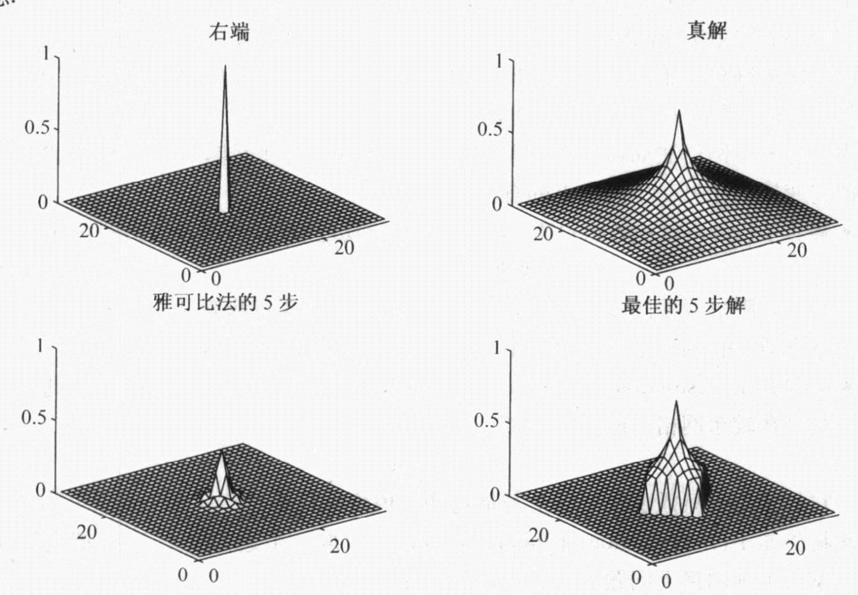


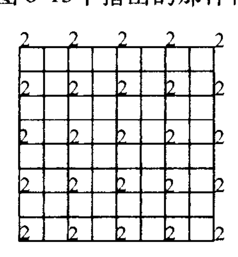
图 6-9 平均相邻的网格点的界限

6.9.1　二维泊松方程多重网格法概述

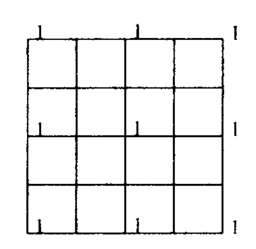
我们先在高层次上叙述算法，然后再补充细节．正如用块循环约化(6.8节)那样，考虑$(2^k-1)\times(2^k-1)$未知量的网格比$2^k\times 2^k$网格有利于更方便地使用FFT(6.7节)．为了理解和执行，在边界上增加已知值为0的节点是方便的，正如在图

332

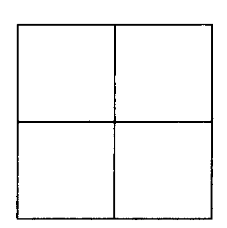
6-10和图6-13中指出的那样得到$(2^k+1)\times(2^k+1)$网格．我们又设$N_k=2^k-1$．



$P^{(3)}$：9×9点的网格
7×7未知量的网格
记号为2的点是下一个较粗的网格的组成部分



$P^{(2)}$：5×5点的网格
3×3未知量的网格
记号为1的点是下一个较粗的网格的组成部分



$P^{(1)}$：3×3点的网格
1×1未知量的网格

图6-10　用二维多重网格法的网格序列

设$P^{(i)}$表示在$(2^i+1)\times(2^i+1)$网格上求解有$(2^i-1)^2$个未知量或等价地在$(N_i+2)\times(N_i+2)$网格上有N_i^2个未知量的离散泊松方程问题．问题$P^{(i)}$由右端$b^{(i)}$和隐含网格大小2^i-1以及系数阵$\boldsymbol{T}^{(i)}\equiv\boldsymbol{T}_{N_i\times N_i}$所确定．$P^{(i)}$的近似解用$x^{(i)}$表示．因而$b^{(i)}$和$x^{(i)}$是在每个网格点上取值的$(2^i-1)\times(2^i-1)$数组．(零边界值是隐含的.)我们将在递增的粗网格上生成有联系的问题$P^{(i)},P^{(i-1)},P^{(i-2)},\cdots,P^{(1)}$的序列，其中$P^{(i-1)}$的解是$P^{(i)}$的解中误差的一个好的近似．

为说明多重网格法如何操作，需要一些在网格上处理问题的算子，不是改进它就是把它变换成在另一个网格上的有关的问题：

- **解算子**(solution operator)S　取问题$P^{(i)}$及其近似解$x^{(i)}$并计算改进的$x^{(i)}$：
$$\text{改进的 } x^{(i)}=S(b^{(i)},x^{(i)}).\tag{6.51}$$
改进是阻尼误差的“高频分量”．在下面将说明这意味着什么．通过作每个网格点值同它的最近相邻值的平均来实施它，这是雅可比法的变形．
- **限制算子**(restriction operator)R　取问题$P^{(i)}$的右端$b^{(i)}$并把它映射到$b^{(i-1)}$，这是在较粗网格上的一个近似：
$$b^{(i-1)}=R(b^{(i)}).\tag{6.52}$$
它的实施也只需要在网格上与最近相邻值作加权平均．
- **插值算子**(interpolation operator)In　对$P^{(i-1)}$取一个近似解$x^{(i-1)}$并把它转换成下一个细网格上问题$P^{(i)}$的近似解$x^{(i)}$：

333

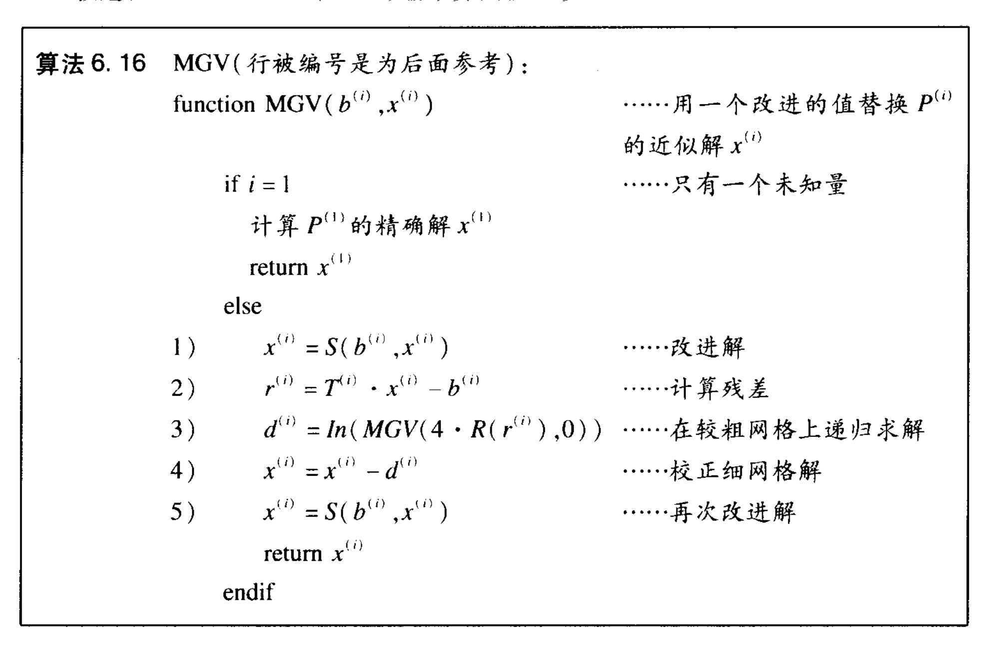
$$x^{(i)}=In(x^{(i-1)}).\tag{6.53}$$

它的实施也只需要在网格上与最近相邻值作加权平均.

因为所有三个算子都用最近相邻值的某个加权平均来替换在每个网格点上的值，所以每种运算对每个未知量，只花费 $O(1)$，或对 n 个未知量花费 $O(n)$. 这就是最终算法低成本的关键.

1. 多重网格 V-循环算法

叙述多重网格 V-循环(MGV)基本算法就足够了.



算法 6.16 MGV(行被编号是为后面参考):

```
function MGV(b^(i), x^(i))                      ……用一个改进的值替换 P^(i) 的近似解 x^(i)
    if i = 1                                    ……只有一个未知量
       计算 P^(1) 的精确解 x^(1)
       return x^(1)
    else
1)        x^(i) = S(b^(i), x^(i))               ……改进解
2)        r^(i) = T^(i) · x^(i) - b^(i)         ……计算残差
3)        d^(i) = In(MGV(4 · R(r^(i)), 0))      ……在较粗网格上递归求解
4)        x^(i) = x^(i) - d^(i)                 ……校正细网格解
5)        x^(i) = S(b^(i), x^(i))               ……再次改进解
          return x^(i)
    endif
```

简言之，算法执行下列工作:

1. 从细网格$(b^{(i)},x^{(i)})$上的一个问题开始.
2. 通过阻尼高频误差改进它: $x^{(i)}=S(b^{(i)},x^{(i)})$.
3. 计算近似解 $x^{(i)}$的残差 $r^{(i)}$.
4. 在下一个较粗的网格上逼近细网格残差 $r^{(i)}:R(r^{(i)})$.
5. 用一个零初始猜测递归地求解较粗的问题: $\mathrm{MGV}(4\cdot R(r^{(i)}),0)$. 因子 4 出现是因为泊松方程右端中的 h^2 因子从细网格到粗网格改变 4 的倍数.
6. 把粗网格解映射回到细网格: $d_i=In(\mathrm{MGV}(4\cdot R(r^{(i)}),0))$.
7. 从细网格解减去在粗网格上计算的校正: $x^{(i)}=x^{(i)}-d^{(i)}$.
8. 更多地改进解: $x^{(i)}=S(b^{(i)},x^{(i)})$.

334

下面简明地证实算法是合理的. (细节在后面讨论). 假如(用归纳法)$d^{(i)}$是方程

$$T^{(i)}\cdot d^{(i)}=r^{(i)}=T^{(i)}\cdot x^{(i)}-b^{(i)}$$

的精确解. 重新安排得到

$$T^{(i)}\cdot(x^{(i)}-d^{(i)})=b^{(i)}$$

所以 $x^{(i)}-d^{(i)}$ 是所求的解.

算法称为V-循环是因为在(网格编号 i，次数)空间中每次递归调用MGV用一个点概略地画出它，它看上去像图6-11那样，在左上角上从调用MGV($b^{(5)},x^{(5)}$)出发. 先对网格4，然后对网格3，等等下降至最粗的网格1，然后再次回到网格5如此调用MGV.

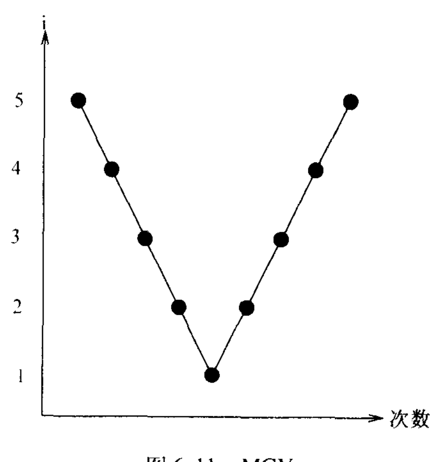


图6-11　MGV

只要知道模块 S，R 和 In 通过网格点的相邻值的某种加权平均来替代网格点上的值，我们做MGV的 $O(\cdot)$ 复杂性分析就足够了. 因为每个模块对每个网格点做固定的工作量，所以它做的总共工作量与网格点数成比例. 因而在V-循环中"V"上的第 i 层网格上的每个点将花费 $O((2^i-1)^2)=O(4^i)$ 次运算. 若最细的网格是在具有 $n=O(4^k)$ 个未知量的第 k 层上，则总共的花费将由下面几何和给出

335

$$\sum_{i=1}^{k} O(4^i)=O(4^k)=O(n).$$

2. 完全的多重网格法

最终的多重网格算法利用刚才描述为一个模块的MGV. 它称为完全的多重网格法(full multigrid，FMG).

算法6.17　FMG:

function FMG($b^{(k)},x^{(k)}$)　……获得 $P^{(k)}$ 的一个精确解 $x^{(k)}$

　　精确地求解 $P^{(1)}$ 得到 $x^{(1)}$

　　for $i=2$ to k

　　　　$x^{(i)}=\text{MGV}(b^{(i)},In(x^{(i-1)}))$

　　end for

简言之，算法做下列工作：

1. 精确地求解最简单的问题 $P^{(1)}$.

2. 给定粗问题 $P^{(i-1)}$ 的一个解 $x^{(i-1)}$，把它映射成下一个较细问题 $P^{(i)}$ 的初始猜测 $x^{(i)}:In(x^{(i-1)})$.

3. 对这个初始猜测利用 MGV 求解较细的问题：MGV$(b^{(i)},In(x^{(i-1)}))$.

现在可做 FMG 的整体 $O(\cdot)$ 复杂性分析. 在图 6-12 中指出在(网格编号 i，次数)空间中 FMG 的一幅图片. 在 FMG 的内循环中每次调用 MGV 在图片中有一个 "V". 如前面一样在第 i 层上出发的 "V" 花费 $O(4^i)$. 因而总共花费再次由下列几何和给出

$$\sum_{i=1}^{k} O(4^i) = O(4^k) = O(n),$$

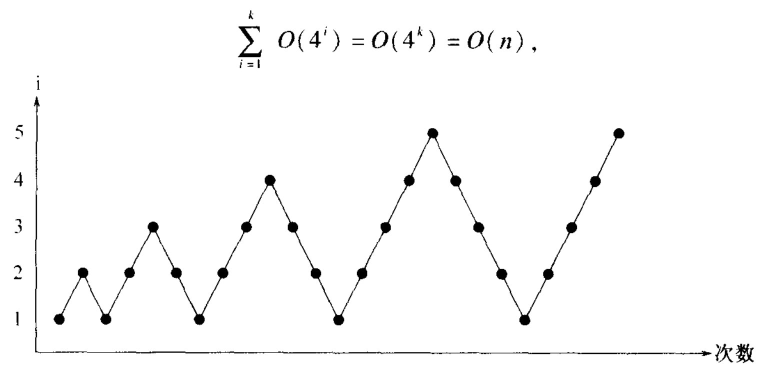


图 6-12 FMG 循环

这是最优的，因为它对每 n 个未知量只做一个固定的工作量. 这就说明了表 6-1 中关于多重网格法的表列数据.

多重网格法的 Matlab 实施过程(关于一维和二维模型问题)在 HOMEPAGE/Matlab/MG_README. html 上可以得到.

336

6.9.2 一维泊松方程的多重网格法详述

现在将详细说明组成多重网格算法的各种算子 S，R 和 In 并概述收敛性证明. 我们将对一维泊松方程讨论这些问题，因为这个方程具备所有有关的特性，但书写比较简单. 特别地，现在可考虑一组嵌套的一维问题代替二维问题，如图 6-13 中所示.

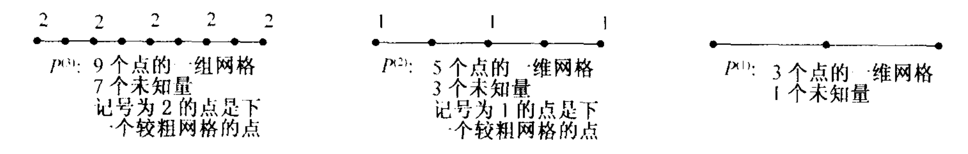


图 6-13 利用一维多重网格法的网格序列

如前面一样用 $P^{(i)}$ 表示网格 i 上求解的问题，即 $T^{(i)}\cdot x^{(i)}=b^{(i)}$，如前面一样 N_i

$=2^i-1$ 以及 $T^{(i)}\equiv T_{N_i}$. 通过描述解算子 S 开始，这是加权雅可比法(weighted Jacobi's method)的形式.

1. 一维解算子

为记号简单起见，在本小节中去掉 $T^{(i)}$, $x^{(i)}$ 和 $b^{(i)}$ 的上标. 如引理6.1中定义的那样设 $T=Z\Lambda Z^{T}$ 是 T 的特征分解，求解 $Tx=b$ 的标准雅可比法是 $x_{m+1}=Rx_m+c$，其中 $R=I-T/2$, $c=b/2$. 我们考虑加权雅可比法 $x_{m+1}=R_w x_m+c_w$，其中 $R_w=I-wT/2$, $c_w=w\,b/2$；$w=1$ 对应于标准的雅可比方法. 注意 $R_w=Z(I-w\Lambda/2)Z^{T}$ 是 R_w 的特征分解. R_w 的特征值以通常的方式确定加权雅可比法的收敛性：设 $e_m=x_m-x$ 是加权雅可比法在 m 次迭代上的误差，所以

$$\begin{aligned} e_m &= R_w e_{m-1} \\ &= R_w^m e_0 \\ &= (Z(I-w\Lambda/2)Z^{T})^m e_0 \\ &= Z(I-w\Lambda/2)^m Z^{T} e_0 \end{aligned}$$

故

$$Z^{T}e_m=(I-w\Lambda/2)^m Z^{T}e_0 \text{ 或 } (Z^{T}e_m)_j=(I-w\Lambda/2)_{jj}^m (Z^{T}e_0)_j.$$

我们称 $(Z^{T}e_m)_j$ 为误差 e_m 的第 j 个频率分量(jth frequency component)，因 $e_m=Z(Z^{T}e_m)$ 是用 $(Z^{T}e_m)_j$ 加权的 Z 的列之和，即不同频率的正弦曲线之和(见图6-2). 特征值 $\lambda_j(R_w)=1-w\lambda_j/2$ 决定每个频率分量如何快速趋向于零. 图6-14对 $N=99$ 和不同的权值 w 画出 $\lambda_j(R_w)$.

337

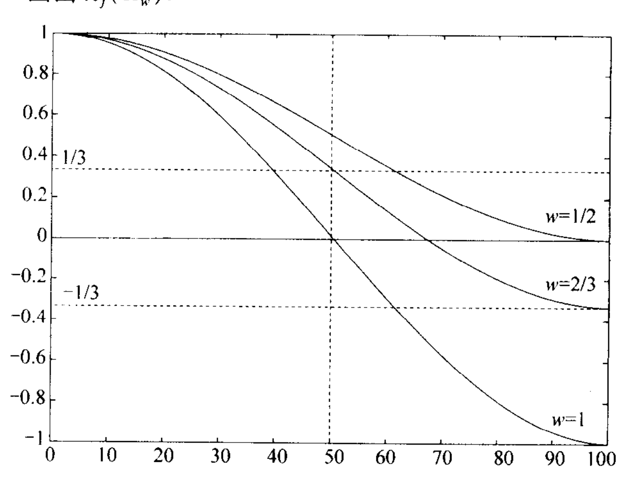


图6-14　对 $N=99$ 和 $w=1$(雅可比法)，$w=1/2$ 和 $w=2/3$ 的 R_w 的谱图像

当 $w=\frac{2}{3}$ 和 $j>\frac{N}{2}$ 时，即关于频率 λ_j 的上面一半有 $|\lambda_j(R_w)|\leqslant\frac{1}{3}$. 这意味着误差分量 $(Z^{T}e_m)_j$ 的上面一半每次迭代乘以 $\frac{1}{3}$ 或小于 $\frac{1}{3}$，与 N 无关. 正如我们将在图

6-15 中看到的，低频误差分量不是同样多地减少. 故用 $w=\frac{2}{3}$的加权雅可比收敛性在减少高频误差方面是好的.

因而，(6.15)式中的解算子 S 由取 $w=\frac{2}{3}$的加权雅可比法收敛的单步组成：

$$S(b,x)=R_{2/3}\cdot x+b/3. \tag{6.54}$$

当我们要指出在网格 i 的 $R_{2/3}$运算时，将改记为 $R_{2/3}^{(i)}$.

图 6-15 指出对 $i=6$ 作 S 的两步的影响，其中有 $2^i-1=63$ 个未知量. 图片有三行，第一行指出初始解和误差，接着两行指出逐次应用 S 之后的解 x_m 和误差 e_m. 真解是一条正弦曲线，在每行最左面的图中用一条虚线表示. 近似解在同样的图中用一条实线表示. 中间的图单独表示误差，包括在底部标出的它的 2－范数. 最右面的图表示误差 Z^Te_m 的频率分量. 在最右面的图中可以看出当应用 S 时，频率分量的右(上)半部分被阻尼. 这也可以在中间和左边的图中看出，因为近似解变得更光滑. 这是因为高频误差看上去像“粗糙的”误差而低频误差看上去像“光滑的”误差. [338] 在起初，向量的范数从 1.65 迅速地减少到 1.055，但是接着慢慢地衰减，因为在高频中有一点点误差阻尼. 因而一次只做几个 S 的迭代，毕竟是有意义的.

2. 多重网格法的递归结构

利用这个术语，我们可描述多重网格法的递归结构如下. 在最细的网格 $P^{(k)}$ 上多重网格法是阻尼解中误差的频率分量之上面一半. 这用刚才描述的解算子 S 来实现. 在下一个具有一半数目点的较粗的网格上，多重网格法阻尼误差中其余的频率分量之上面一半. 这是因为取一个具有一半数目点的较粗的网格，使得频率作为高频出现二次，如下例中说明的一样.

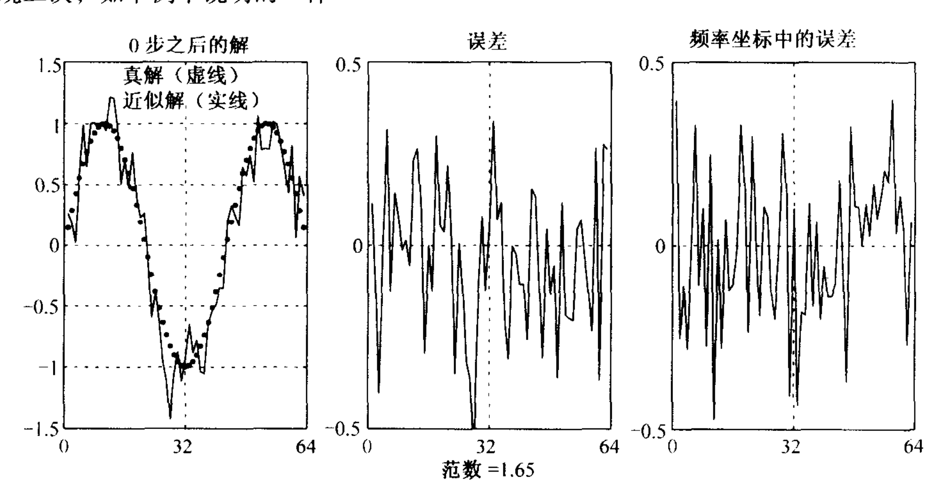


图 6-15 加权雅可比收敛性的说明

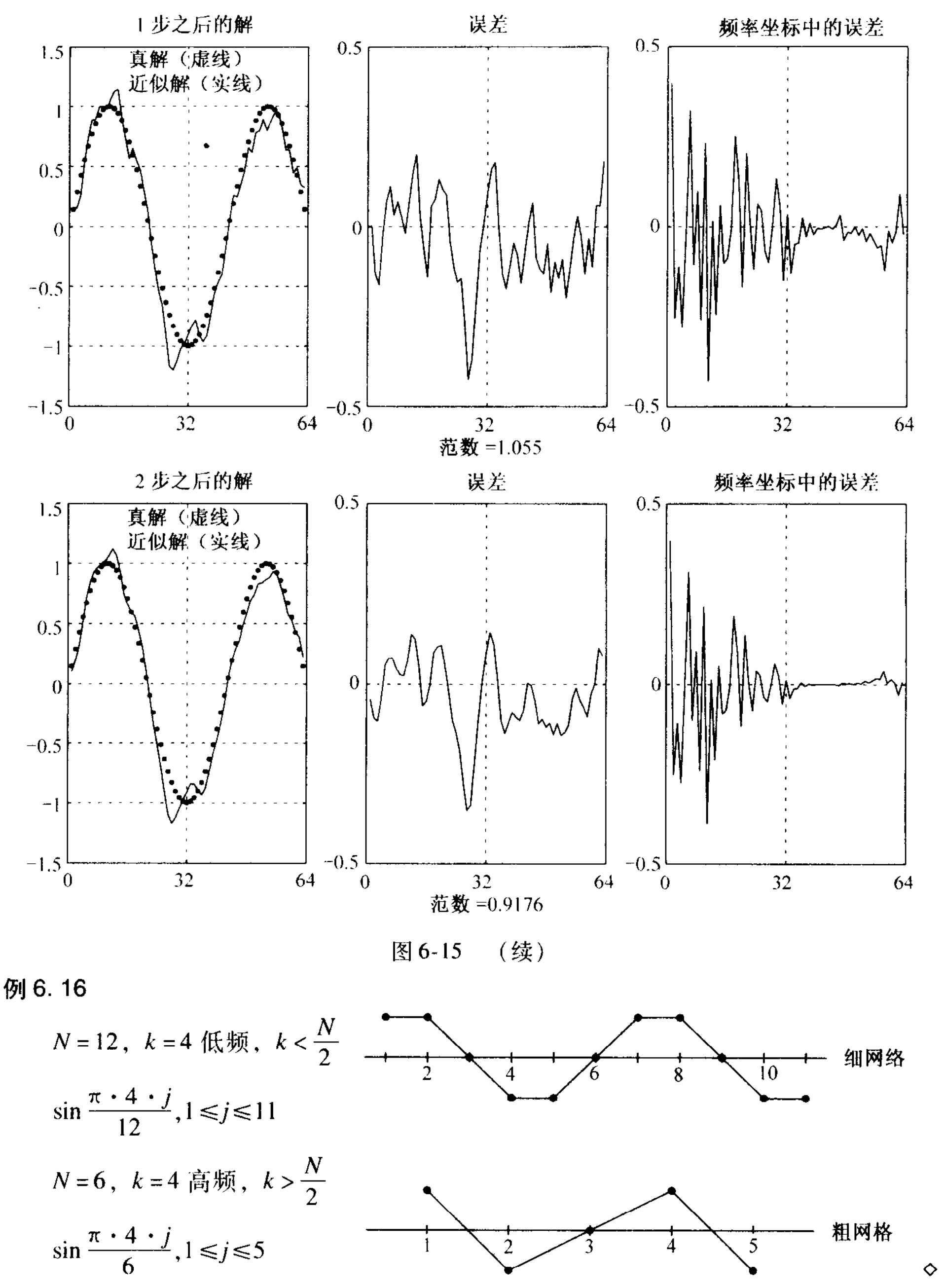


图6-15　（续）

例 6.16

$N=12$，$k=4$ 低频，$k<\frac{N}{2}$

$\sin\frac{\pi\cdot 4\cdot j}{12}, 1\leqslant j\leqslant 11$

$N=6$，$k=4$ 高频，$k>\frac{N}{2}$

$\sin\frac{\pi\cdot 4\cdot j}{6}, 1\leqslant j\leqslant 5$

◇

在下一个较粗网格上，阻尼其余的频率分量的上面一半. 如此等等，直到我们求解精确的(一个未知量)问题 $P^{(1)}$. 这在图6-16 中概略地看出. 限制算子和插值算子的目的是把一个网格点上的近似解变换到下一个较粗的网格上或下一个较细的网格上.

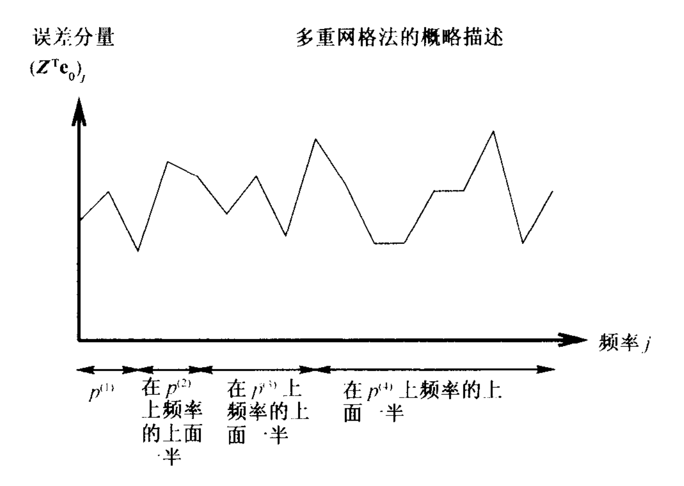


图 6-16 多重网格法如何阻尼误差分量的概略描述

3. 一维限制算子

现在转到限制算子 R，它从问题 $P^{(i)}$ 中取一个右端 $r^{(i)}$ 并在下一个粗网格上逼近它，得到 $r^{(i-1)}$.

计算 $r^{(i-1)}$ 最简单的方法应该是在粗网格和细网格的公共网格点上简单的抽样(sample) $r^{(i)}$. 但是较好的办法是在相邻的细网格点上用 $r^{(i)}$ 的平均值计算在粗网格上的 $r^{(i-1)}$：在粗网格上的值是 0.5 乘相应的细网格点上的值，加 0.25 乘每个细网格点相邻点上的值. 我们称这个为平均. 两个方法都在图 6-17 中说明.

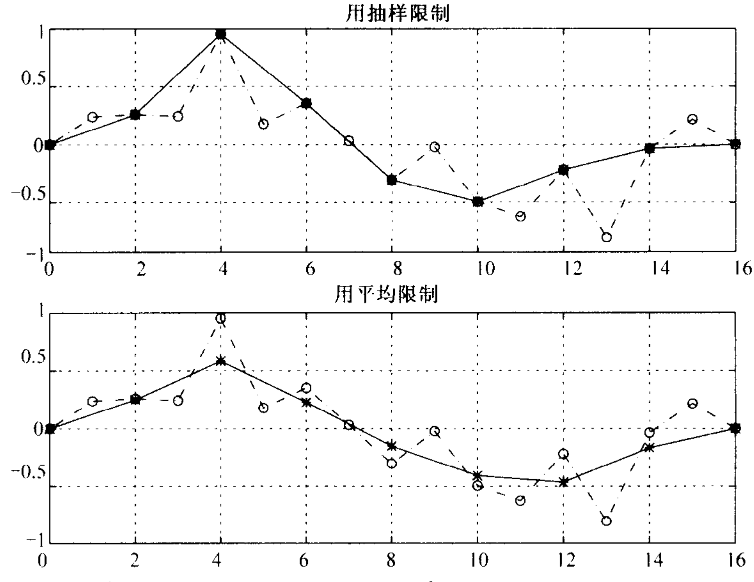


图 6-17 把具有 $2^4-1=15$ 个点的网格限制到具有 $2^3-1=7$ 个点的网格上(也指出 0 边界值.)

总之，记限制运算为

$$
\begin{aligned}
r^{(i-1)} &= R(r^{(i)}) \\
&\equiv P_i^{i-1} \cdot r^{(i)} \\
&= \begin{bmatrix}
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & & & & \\
 & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & & \\
 & & & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & \\
 & & & & & \ddots & \ddots & \ddots & \\
 & & & & & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4}
\end{bmatrix} \cdot r^{(i)}.
\end{aligned} \tag{6.55}
$$

在矩阵 P_i^{i-1} 上的下标 i 和上标 $i-1$ 表示它把具有 2^i-1 个点的网格映射到具有$2^{i-1}-1$ 个点的网格上.

在二维情况，限制涉及作每个网格点的 8 个最接近的相邻点的平均：1/4 乘网格单元本身的值，加 1/8 乘左、右、上和下四个相邻点上的值，加 1/16 乘左上、左下、右上和右下四个其余相邻点上的值.

4. 一维插值算子

插值算子 In 在粗网格上取一个近似解 $d^{(i-1)}$ 并把它映射到下一个较细网格上的一个函数 $d^{(i)}$. 解 $d^{(i-1)}$ 被插值到较细的网格如图 6-18 所示：做简单的线性组合填补到细网格上的值中.（利用边界值已知为零的事实）. 数学上，记这些为

$$
d^{(i)} = In(d^{(i-1)}) \equiv P_{i-1}^{i} \cdot d^{(i-1)} = \begin{bmatrix}
\frac{1}{2} & & & \\
1 & & & \\
\frac{1}{2} & \frac{1}{2} & & \\
 & 1 & \ddots & \\
 & \frac{1}{2} & \ddots & \frac{1}{2} \\
 & & \ddots & 1 \\
 & & & \frac{1}{2}
\end{bmatrix} \cdot d^{(i-1)}. \tag{6.56}
$$

矩阵 P_{i-1}^{i} 上的下标 $i-1$ 和上标 i 表示它把具有 $2^{i-1}-1$ 个点的网格映射到具有 2^i-1 个点的网格.

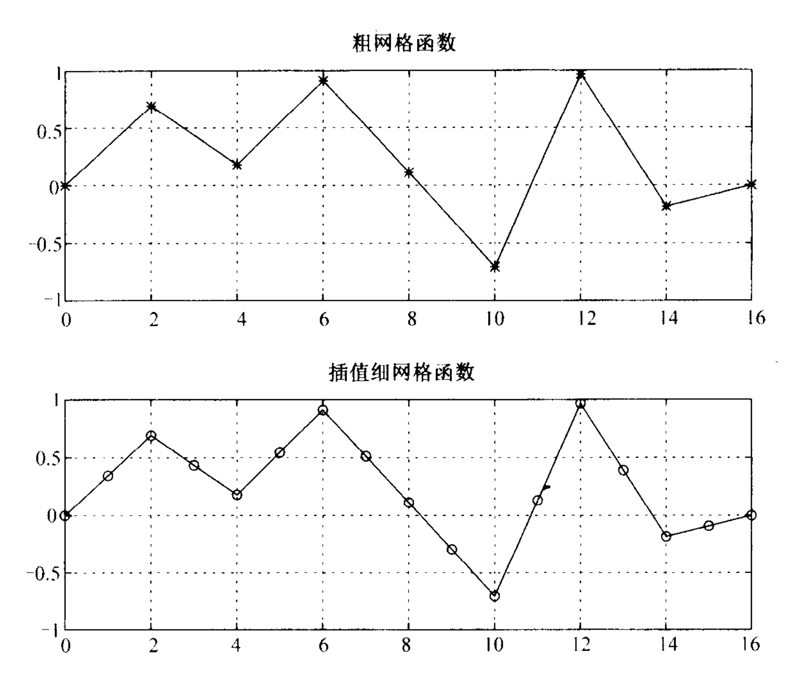


图6-18 把具有 $2^3-1=7$ 个点的网格插值到具有 $2^4-1=15$ 个点的网格上(也指出0边界值.)

注意 $P_{i-1}^{i}=2\cdot(P_{i}^{i-1})^{\mathrm{T}}$. 换言之，插值和光滑本质上彼此互为转置. 这个事实在后面的收敛性分析中是重要的.

在二维情形，插值还涉及作一个细网格点的最近的粗相邻点上的值的平均.（若细网格点也是一个粗网格点，则算一个相邻点；若细网格点的最接近的粗相邻点是左和右或上和下，则算二个相邻点；否则算四个相邻点).

5. 把它放在一起

现在对图6-19上面两个图中描述的问题运行刚才描述的算法8次迭代；同时指出真解 x(在左上方)和右端 b(在右上方). 未知量的个数是 $2^7-1=127$. 我们指出在底部的三个图中多重网格法如何收敛. 中间左边的图指出连贯的残差之比 $\|r_{m+1}\|/\|r_m\|$，其中下标 m 是多重网格法(即调用FMG或算法6.17)的迭代次数. 这些比约为 .15，表示每次多重网格迭代残差减少了超过6倍. 在中间右边的图中指出这个快速收敛性，它表示 $\|r_m\|$ 对 m 的半对数图. 正如预期的那样它是一条斜率为 $\log_{10}(.15)$ 的直线. 最后，底部的图画出全部八个误差向量 x_m-x. 我们在半对数图上看出如何使它们光顺并且变成平行，在相邻的 $\log_{10}(.15)$ 的图之间以一个常数减小.

图6-20指出二维模型问题的一个类似的例子.

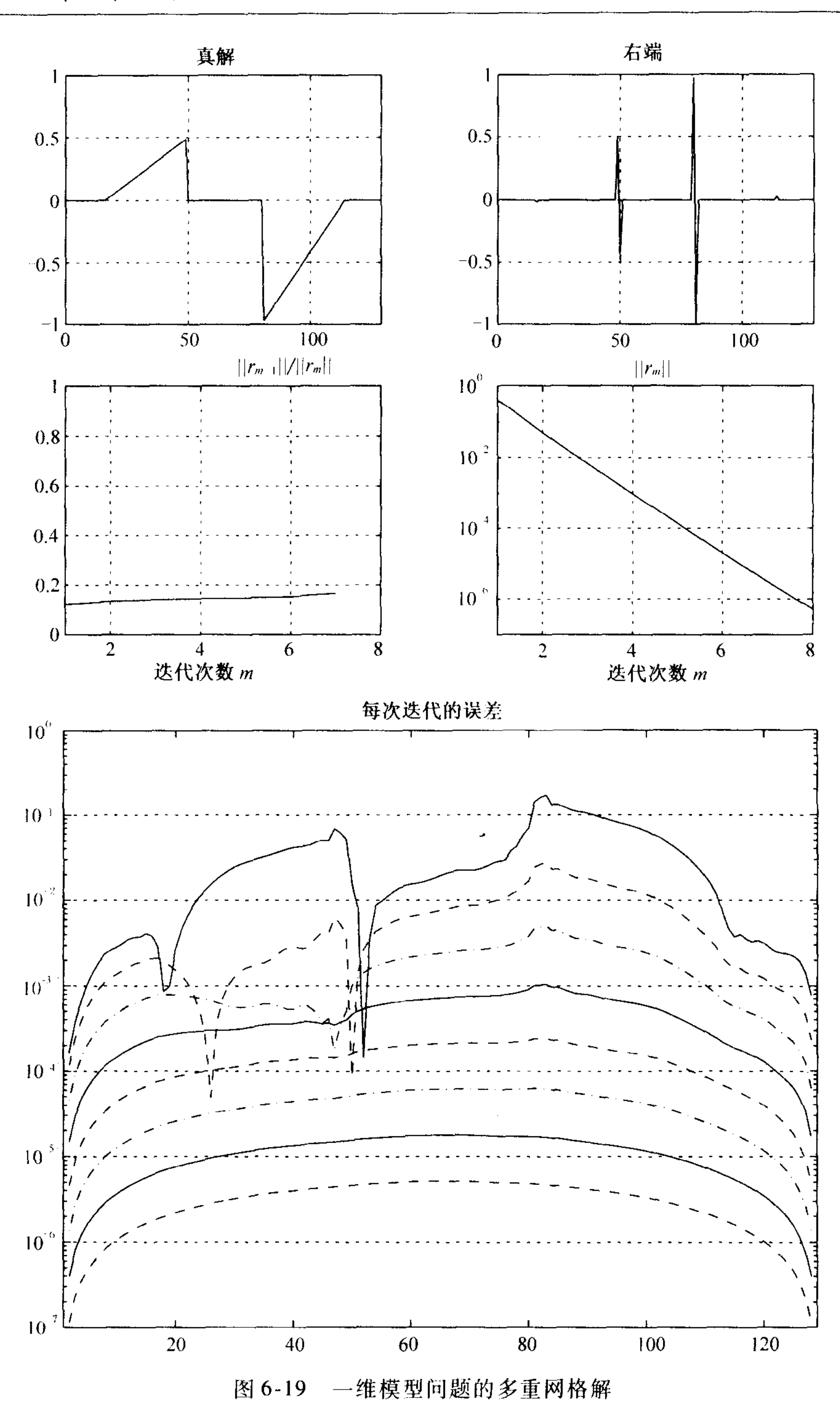


图6-19　一维模型问题的多重网格解

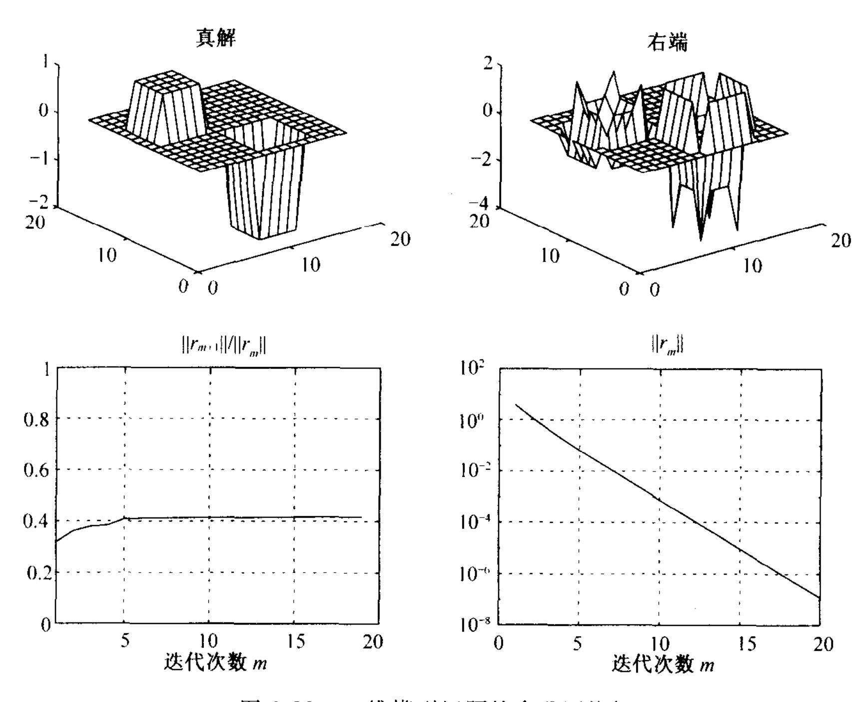


图 6-20 二维模型问题的多重网格解

6. 收敛性证明

最后，我们概述收敛性证明，指出在一个 FMG “V” - 循环中总的误差减少了小于 1 的常数倍，与网格大小 $N_k = 2^k - 1$ 无关. 这意味着为了以任何小于 1 的倍数来减少误差所需的 FMG V-循环次数与 k 无关，故全部的工作与单个 FMG V-循环的代价成比例，即与未知量 n 的个数成比例.

我们将弄清单个 V-循环来简化证明并且归纳假定粗网格问题是精确地求解的[43]. 实际上粗网格问题不是十分精确地求解的，但这个粗糙的分析足够具备证明的精神实质：低频误差在较粗的网格上被减弱而高频误差在细网格上被消除.

现在书写所有定义 V-循环的公式并把它们全部组合成形式为 “新 $\boldsymbol{e}^{(i)} = \boldsymbol{M} \cdot \boldsymbol{e}^{(i)}$” 的单独的公式，其中 $\boldsymbol{e}^{(i)} = \boldsymbol{x}^{(i)} - \boldsymbol{x}$ 是误差而 $\boldsymbol{M}$ 是一个矩阵，它的特征值决定收敛率；我们的目标是证明它们被界定远离 1，与 i 无关. 在下列表中的行数参见算法 6.16.

339 ~ 345

(a) $\boldsymbol{x}^{(i)} = S(b(i), x(i)) = \boldsymbol{R}_{2/3}^{(i)}\boldsymbol{x}^{(i)} + \boldsymbol{b}^{(i)}/3$ 由第 1)行和(6.54)式

(b) $\boldsymbol{r}^{(i)} = \boldsymbol{T}^{(i)} \cdot \boldsymbol{x}^{(i)} - \boldsymbol{b}^{(i)}$ 由第 2)行

$\boldsymbol{d}^{(i)} = In(MGV(4 \cdot \boldsymbol{R}(\boldsymbol{r}^{(i)}), 0))$ 由第 3)行

$= In([\boldsymbol{T}^{(i-1)}]^{-1}(4 \cdot \boldsymbol{R}(\boldsymbol{r}^{(i)})))$ 由假设粗网格问题是精确求解的

$= In([\boldsymbol{T}^{(i-1)}]^{-1}(4 \cdot \boldsymbol{P}_i^{i-1}\boldsymbol{r}^{(i)}))$ 由(6.55)式

(c) $\quad = P_{i-1}^{i}([T^{(i-1)}]^{-1}(4 \cdot P_{i}^{i-1} r^{(i)}))$ 　　由(6.56)式

(d) $x^{(i)} = x^{(i)} - d^{(i)}$ 　　由第4)行

(e) $x^{(i)} = S(b(i), x(i)) = R_{2/3}^{(i)} x^{(i)} + b^{(i)}/3$ 　　由第5)行

为了得到更新误差 $e^{(i)}$ 的式子，从上面的(a)行和(e)行中减去恒等式 $x = R_{2/3}^{(i)} x + b^{(i)}/3$，从(b)行减去 $0 = T^{(i)} \cdot x - b^{(i)}$ 以及从（d）行减去 $x = x$ 得到

(a) $e^{(i)} = R_{2/3}^{(i)} e^{(i)}$,

(b) $r^{(i)} = T^{(i)} \cdot e^{(i)}$,

(c) $d^{(i)} = P_{i-1}^{i}([T^{(i-1)}]^{-1}(4 \cdot P_{i}^{i-1} r^{(i)}))$,

(d) $e^{(i)} = e^{(i)} - d^{(i)}$,

(c) $e^{(i)} = R_{2/3}^{(i)} e^{(i)}$.

把上面每个等式代入到下一个等式得到下列公式，表明如何用一个 V-循环更新误差：

$$\text{新}\, e^{(i)} = R_{2/3}^{(i)} \{ I - P_{i-1}^{i} \cdot [T^{(i-1)}]^{-1} \cdot (4 \cdot P_{i}^{i-1} T^{(i)}) \} R_{2/3}^{(i)} \cdot e^{(i)} \equiv M \cdot e^{(i)}. \tag{6.57}$$

现在需要计算 M 的特征值. 首先利用 $P_{i-1}^{i} = 2 \cdot (P_{i}^{i-1})^{\mathrm{T}}$ 和

$$T^{(i-1)} = 4 \cdot P_{i}^{i-1} T^{(i)} P_{i-1}^{i} = 8 \cdot P_{i}^{i-1} T^{(i)} (P_{i}^{i-1})^{\mathrm{T}} \tag{6.58}$$

简化(6.57)式(见问题6.15). 把这些代入到(6.57)中 M 的表达式得到

$$M = R_{2/3}^{(i)} \{ I - (P_{i}^{i-1})^{\mathrm{T}} \cdot [P_{i}^{i-1} T^{(i)} (P_{i}^{i-1})^{\mathrm{T}}]^{-1} \cdot (P_{i}^{i-1} T^{(i)}) \} R_{2/3}^{(i)}$$

或者，去掉指标以简化记号，

$$M = R_{2/3} \{ I - P^{\mathrm{T}} \cdot [PTP^{\mathrm{T}}]^{-1} \cdot PT \} R_{2/3}. \tag{6.59}$$

继续下去利用所有组成 M 的矩阵($T, R_{2/3}$ 和 P)可分别地利用 $T = T^{(i)}$ 和 $T^{(i-1)}$ 的特征向量矩阵 $Z = Z^{(i)}$ 和 $Z^{(i-1)}$（几乎）对角化：记住 $Z = Z^{\mathrm{T}} = Z^{-1}, T = Z\Lambda Z$ 和 $R_{2/3} = Z(I - \Lambda/3)Z \equiv Z\Lambda_R Z$. 我们把它留给读者去证实 $Z^{(i-1)} P Z^{(i)} = \Lambda_P$，其中 Λ_P 是几乎对角阵（见问题6.15）：

346

$$\lambda_{P,jk} = \begin{cases} \left(+1 + \cos \dfrac{\pi j}{2^i}\right)/\sqrt{8} & \text{若 } k = j \\ \left(-1 + \cos \dfrac{\pi j}{2^i}\right)/\sqrt{8} & \text{若 } k = 2^i - j \\ 0 & \text{否则} \end{cases} \tag{6.60}$$

这允许我们记

$$\begin{aligned} ZMZ &= (ZR_{2/3}Z)\{ I - (ZP^{\mathrm{T}} Z^{(i-1)}) \cdot [(Z^{(i-1)} PZ)(ZTZ)(ZP^{\mathrm{T}} Z^{(i-1)})]^{-1} \\ &\quad \cdot (Z^{(i-1)} PZ)(ZTZ) \} \cdot (ZR_{2/3}Z) \\ &= \Lambda_R \cdot \{ I - \Lambda_P^{\mathrm{T}} [\Lambda_P \Lambda \Lambda_P^{\mathrm{T}}]^{-1} \Lambda_P \Lambda \} \cdot \Lambda_R. \end{aligned}$$

因为 $Z = Z^{-1}$，所以矩阵 ZMZ 与 M 相似，故它与 M 有相同的特征值. 此外，ZMZ 是几乎对角阵：它仅在其主对角线和“副对角线”（从矩阵左下角到右上角的对角线）

上有非零元. 这允许我们明确地计算 M 的特征值.

定理6.11 矩阵 M 有与 i 无关的特征值 1/9 和 0. 所以, 多重网格法以一个与未知量个数无关的固定速率收敛.

证明见问题6.15. 更一般的分析见[268].

这个算法的执行过程见问题6.16. 网站[91]含有一个广泛的文献、软件等等内容的指示性信息.

6.10 区域分解法

求解稀疏线性方程组的区域分解法是一个当前的研究课题, 最新的综述是[49,116,205], 特别是[232]. 我们将仅给出几个简单的例子.

由实际问题的不规则和大小而引起, 也由并行计算机需要的算法而产生的这些方法超出我们曾讨论过的那些方法的范围. 迄今为止我们讨论的最快的方法, 即那些基于块循环约化、FFT 和多重网格的方法, 对(或仅对)诸如模型问题那样的特别规则的问题, 即对长方形上均匀网格离散的泊松方程的效果最佳. 但实际问题的解区域可能不是长方形, 而是不规则的形状(如机翼状,见图 2-12). 图 2-12 也说明有可能区域中预期光滑性较差解的网格点比区域中具有光滑解的网格点多. 此外可能有比泊松方程更复杂的微分方程, 或者甚至在不同区域中有不同的方程. 与问题是否规则无关, 它可能太大以致不能存放在计算机存储器中, 而不得不“分片”求解, 或者试图把问题拆成可以在并行计算机上并行求解的片. 347

区域分解通过指出如何从前面几节中讨论过的较简单的方法系统地创造“混合”方法, 来处理所有这些问题. 这些较简单的方法被应用于总体问题中较小和较规则的子问题, 之后“拼合”这些部分解得到总体解. 若整个问题不能存放在存储器中或者不能在并行计算机上并行求解, 则可以一次求解一个子问题. 下面给出一些例子. 一般有许多方法把一个大问题拆开成片, 也有许多方法求解单独的片, 并且有许多方法把片的解拼合在一起. 区域分解理论不是一种可从一切情况中选择最佳方法的灵丹妙药, 它只是提出了一组可供尝试的合理方法. 在某些情况下(诸如十分像泊松方程那样的问题), 这个理论得到“最佳方法”(每个未知量花费 $O(1)$ 工作量).

我们把讨论分成两部分, *无交叠方法*(nonoverlapping method)和*交叠方法*(overlapping method).

6.10.1 无交叠方法

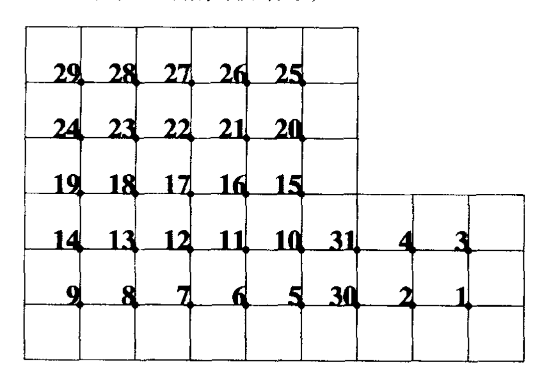
这个方法在文献中也称为*子结构法*(substructuring)或*舒尔补方法*. 它把大问题拆成能存放在计算机存储器中较小的问题, 这个方法在结构分析界中已使用了数十年.

为简单起见, 将利用5-点格式但不是正方形区域而是在 L-*形状区域*上离散通常的带狄利克雷边界条件的泊松方程来说明这个方法. 这个区域可分解成两个区域: 一个小

的正方形和一个两倍边长的大正方形，其中小的正方形与较大的正方形的右边底部相连. 我们将设计一个可以在正方形区域上快速求解问题的解算器.

348

在下图中，关于粗离散化指出每个网格点的编号（编号是在上面的并在对应的网格点的左面；只有“L”内部的网格点被编号）.



注意首先把两个子区域内部的网格点编号（1 到 4 和 5 到 29），然后对边界上的网格点编号（30 和 31）. 所得的矩阵为

4	-1	-1																												
-1	4		-1																										-1	
-1		4	-1																											
	-1	-1	4																											-1
				4	-1				-1																				-1	
				-1	4	-1				-1																				
					-1	4	-1				-1																			
						-1	4	-1				-1																		
							-1	4					-1																	
				-1					4	-1				-1																-1
					-1				-1	4	-1				-1															
						-1				-1	4	-1				-1														
							-1				-1	4	-1				-1													
								-1				-1	4					-1												
									-1					4	-1				-1											
										-1				-1	4	-1				-1										
											-1				-1	4	-1				-1									
												-1				-1	4	-1				-1								
													-1				-1	4					-1							
														-1					4	-1				-1						
															-1				-1	4	-1				-1					
																-1				-1	4	-1				-1				
																	-1				-1	4	-1				-1			
																		-1				-1	4					-1		
																			-1					4	-1					
																				-1				-1	4	-1				
																					-1				-1	4	-1			
																						-1				-1	4	-1		
																							-1				-1	4		
	-1			-1																									4	-1
			-1						-1																				-1	4

$$
\equiv A \equiv \left[\begin{array}{c||c||c} A_{11} & 0 & A_{13} \\ \hline\hline 0 & A_{22} & A_{23} \\ \hline\hline A_{13}^{\mathrm{T}} & A_{23}^{\mathrm{T}} & A_{33} \end{array}\right].
$$

349

这里$A_{11}=T_{2\times 2}$，$A_{22}=T_{5\times 5}$，$A_{33}=T_{2\times 1}\equiv T_2+2I_2$，其中$T_N$由(6.3)式定义，$T_{N\times N}$由(6.14)式定义. 这个矩阵最重要的性质之一是$A_{12}=0$，因为两个子区域内部的网格点不存在直接的耦合. 仅有的耦合穿过边界，它被最后编号(网格点30和31). 因而A_{13}含有最小的正方形和边界之间的耦合，而A_{23}包含大的正方形和边界的耦合.

为观察如何利用A的特殊结构的优点求解$Ax=b$，记A的块LDU分解如下：

$$
A=\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ A_{13}^{\mathrm{T}}A_{11}^{-1} & A_{23}^{\mathrm{T}}A_{22}^{-1} & I \end{bmatrix}\cdot\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & S \end{bmatrix}\cdot\begin{bmatrix} A_{11} & 0 & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & 0 & I \end{bmatrix},
$$

其中

$$
S=A_{33}-A_{13}^{\mathrm{T}}A_{11}^{-1}A_{13}-A_{23}^{\mathrm{T}}A_{22}^{-1}A_{23} \tag{6.61}
$$

称为包含A_{11}和A_{22}的前主子矩阵的舒尔补. 所以可记

$$
A^{-1}=\begin{bmatrix} A_{11}^{-1} & 0 & -A_{11}^{-1}A_{13} \\ 0 & A_{22}^{-1} & -A_{22}^{-1}A_{23} \\ 0 & 0 & I \end{bmatrix}\cdot\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & S^{-1} \end{bmatrix}\cdot\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -A_{13}^{\mathrm{T}}A_{11}^{-1} & -A_{23}^{\mathrm{T}}A_{22}^{-1} & I \end{bmatrix}.
$$

因此为了用A^{-1}乘一个向量，必须用A^{-1}的这个分解形式中的块，即A_{13}和A_{23}(及其转置)，A_{11}^{-1}，A_{22}^{-1}和S^{-1}相乘. 用A_{13}和A_{23}相乘是便宜的，因为它们非常稀疏，用A_{11}^{-1}和A_{22}^{-1}相乘也是便宜的，因为我们选择这些子区域可利用FFT块循环约化，多重网格以及一些迄今为止讨论过的其他快速方法求解. 余下说明如何用S^{-1}相乘.

因为边界上的网格点比子区域中的网格点少得多，所以A_{33}和S的维数比A_{11}和A_{22}的维数小得多；对较细的网格间距这个影响增大. S像A一样是对称正定阵，而且(在此情况下)是稠密的. 为明确地计算它我们需要关于每个子区域对每个边界网格点求解一次(从(6.61)中的$A_{11}^{-1}A_{13}$和$A_{22}^{-1}A_{23}$项). 这当然能操作，之后可利用稠密楚列斯基分解，分解S并继续下去求解方程组. 但这样做比只用S乘一个向量代价昂贵得多，后者只需要利用(6.61)式对每个子区域求解一次. 这就使得基于克雷洛夫子空间的迭代法例如CG显得有吸引力(6.6节)，因为这些方法只需要用S乘一个向量. CG需要的矩阵-向量乘法次数依赖于S的条件数. 使区域分解如此有吸引力的本质是S比原来的矩阵A有较好的条件(条件数逐渐变成像$O(N)$而不是$O(N^2)$)，故收敛迅速[116,205].

350

更一般地，我们有边界隔离的 $k>2$ 个子区域(见图6.21，其中粗黑线隔离子区域)．若把每个子区域中的节点连贯地编号，随后边界节点编号，得到矩阵

$$A=\left[\begin{array}{ccc|c} A_{1,1} & & \mathbf{0} & A_{1,k+1} \\ & \ddots & & \vdots \\ \mathbf{0} & & A_{k,k} & A_{k,k+1} \\ \hline A_{1,k+1}^{\mathrm{T}} & \cdots & A_{k,k+1}^{\mathrm{T}} & A_{k+1,k+1} \end{array}\right], \tag{6.62}$$

通过独立地分解每个 $A_{i,i}$ 可以再分解 A 并构成舒尔补 $S=A_{k+1,k+1}-\sum_{i=1}^{k}A_{i,k+1}^{\mathrm{T}}A_{i,i}^{-1}A_{i,k+1}$.

此时，当存在多于一个边界线段时，S 具有进一步可被用来对它预处理的结构．例如，在边界线段交集上的网格点之前先对每个边界线段内部的网格点编号，我们得到如 A 中那样的结构．S 的对角块是复杂的，但是它可用 $T_N^{1/2}$ 近似，这可用 FFT 有效地转化[36,37,38,39,40]．总结目前发展水平，通过适当地选择 S 的预条件子，我们可使 CG 的步数与边界网格点的个数 N 无关[231].

6.10.2 交叠方法

上节中的方法称为无交叠的；因为对应于 $A_{i,i}$ 中的节点区域不相交，所以导致(6.62)式中的块对角结构．在本节中允许如下图中指出的那样的交叠区域．正如我们将看到的那样，这个交叠允许我们设计一个与多重网格法速度相当的但可应用于更广泛一组问题的算法.

在图中具有虚线边界的长方形是区域 Ω_1，而具有实线边界的正方形是区域 Ω_2. 351 我们已重新对节点编号，首先对 Ω_1 中的节点编号，最后对 Ω_2 中的节点编号，而在交叠 $\Omega_1\cap\Omega_2$ 中的节点在中间编号

Ω_2
31 26 21 16 13
30 25 20 15 12
29 24 19 14 11
28 23 18 10 8 6 4 2
27 22 17 9 7 5 3 1
Ω_1

在下面的矩阵 A 中显示了这些区域，这个矩阵与6.10.1节中的一样，但它的行和列的次序如上面所示：

$$
\left[\begin{array}{cccccc||cccc|ccccccccccccccccccccc}
4&-1&-1&&&&&&&&&&&&&&&&&&&&&&&&&&&&\\
-1&4&&-1&&&&&&&&&&&&&&&&&&&&&&&&&&&\\
-1&&4&-1&-1&&&&&&&&&&&&&&&&&&&&&&&&&&\\
&-1&-1&4&&-1&&&&&&&&&&&&&&&&&&&&&&&&&\\
&&-1&&4&-1&-1&&&&&&&&&&&&&&&&&&&&&&&&\\
&&&-1&-1&4&&-1&&&&&&&&&&&&&&&&&&&&&&&\\
\hline\hline
&&&&-1&&4&-1&-1&&&&&&&&&&&&&&&&&&&&&&\\
&&&&&-1&-1&4&&-1&-1&&&&&&&&&&&&&&&&&&&&\\
&&&&&&-1&&4&-1&&&&&&&-1&&&&&&&&&&&&&&\\
&&&&&&&-1&-1&4&&&&-1&&&&-1&&&&&&&&&&&&&\\
\hline
&&&&&&&-1&&&4&-1&&-1&&&&&&&&&&&&&&&&&\\
&&&&&&&&&&-1&4&-1&&-1&&&&&&&&&&&&&&&&\\
&&&&&&&&&&&-1&4&&&-1&&&&&&&&&&&&&&&\\
&&&&&&&&&-1&-1&&&4&-1&&&&-1&&&&&&&&&&&&\\
&&&&&&&&&&&-1&&-1&4&-1&&&&-1&&&&&&&&&&&\\
&&&&&&&&&&&&-1&&-1&4&&&&&-1&&&&&&&&&&\\
&&&&&&&&-1&&&&&&&&4&-1&&&&-1&&&&&&&&&\\
&&&&&&&&&-1&&&&&&&-1&4&-1&&&&-1&&&&&&&&\\
&&&&&&&&&&&&&-1&&&&-1&4&-1&&&&-1&&&&&&&\\
&&&&&&&&&&&&&&-1&&&&-1&4&-1&&&&-1&&&&&&\\
&&&&&&&&&&&&&&&-1&&&&-1&4&&&&&-1&&&&&\\
&&&&&&&&&&&&&&&&-1&&&&&4&-1&&&&-1&&&&\\
&&&&&&&&&&&&&&&&&-1&&&&-1&4&-1&&&&-1&&&\\
&&&&&&&&&&&&&&&&&&-1&&&&-1&4&-1&&&&-1&&\\
&&&&&&&&&&&&&&&&&&&-1&&&&-1&4&-1&&&&-1&\\
&&&&&&&&&&&&&&&&&&&&-1&&&&-1&4&&&&&-1\\
&&&&&&&&&&&&&&&&&&&&&-1&&&&&4&-1&&&\\
&&&&&&&&&&&&&&&&&&&&&&-1&&&&-1&4&-1&&\\
&&&&&&&&&&&&&&&&&&&&&&&-1&&&&-1&4&-1&\\
&&&&&&&&&&&&&&&&&&&&&&&&-1&&&&-1&4&-1\\
&&&&&&&&&&&&&&&&&&&&&&&&&-1&&&&-1&4
\end{array}\right]
$$

我们用对矩阵进行分块的方法指出区域之间的边界：单线把矩阵分成与 Ω_1 相联系的节点(1 到 10)和与 $\Omega\backslash\Omega_1$ 相联系的其余节点(11 到 31)．双线把矩阵分成与 Ω_2 相联系的节点．(7 到 31)和与 $\Omega\backslash\Omega_2$ 相联系的节点(1 到 6)．下面的子矩阵据此标在下角上： 352

$$
\boldsymbol{A}=\left[\begin{array}{c|c}
\boldsymbol{A}_{\Omega_1,\Omega_1} & \boldsymbol{A}_{\Omega_1,\Omega\backslash\Omega_1} \\
\hline
\boldsymbol{A}_{\Omega\backslash\Omega_1,\Omega_1} & \boldsymbol{A}_{\Omega\backslash\Omega_1,\Omega\backslash\Omega_1}
\end{array}\right]=\left[\begin{array}{c||c}
\boldsymbol{A}_{\Omega\backslash\Omega_2,\Omega\backslash\Omega_2} & \boldsymbol{A}_{\Omega\backslash\Omega_2,\Omega_2} \\
\hline\hline
\boldsymbol{A}_{\Omega_2,\Omega\backslash\Omega_2} & \boldsymbol{A}_{\Omega_2,\Omega_2}
\end{array}\right].
$$

相应地，分隔向量为

$$
\begin{aligned}
\boldsymbol{x} &= \left[\begin{array}{c}
\boldsymbol{x}_{\Omega_1} \\
\hline
\boldsymbol{x}_{\Omega\backslash\Omega_1}
\end{array}\right]=\left[\begin{array}{c}
\boldsymbol{x}\ (1:10) \\
\hline
\boldsymbol{x}\ (11:31)
\end{array}\right] \\
&= \left[\begin{array}{c}
\boldsymbol{x}_{\Omega\backslash\Omega_2} \\
\hline\hline
\boldsymbol{x}_{\Omega_2}
\end{array}\right]=\left[\begin{array}{c}
\boldsymbol{x}\ (1:6) \\
\hline\hline
\boldsymbol{x}\ (7:31)
\end{array}\right].
\end{aligned}
$$

现在已有足够的记号去叙述两个基本的交叠的区域分解算法，由于历史原因，最简单的一个称为加性施瓦茨方法(additive schwarz method)．但因为它与 6.6 和

6.6.5节的(块)雅可比迭代的相似性，也被称为交叠的块雅可比迭代(overlapping block Jacobi iteration).

算法6.18　更新 $Ax=b$ 的近似解 x_i 得到一个较好的解 x_{i+1} 的加性施瓦茨方法：

$r=b-Ax_i$　/* 计算残差 */

$x_{i+1}=0$

$x_{i+1,\Omega_1}=x_{i,\Omega_1}+A^{-1}_{\Omega_1,\Omega_1}\cdot r_{\Omega_1}$　/* 更新 Ω_1 上的解 */

$x_{i+1,\Omega_2}=x_{i+1,\Omega_2}+A^{-1}_{\Omega_2,\Omega_2}\cdot r_{\Omega_2}$　/* 更新 Ω_2 上的解 */

这个算法也可写成单行

$$x_{i+1}=x_i+\left[\frac{A^{-1}_{\Omega_1,\Omega_1}\cdot r_{\Omega_1}}{0}\right]+\left[\frac{0}{A^{-1}_{\Omega_2,\Omega_2}\cdot r_{\Omega_2}}\right].$$

简言之，算法操作如下：利用依赖于前面的近似解 x_i 在节点11，14，17，18和19上的边界条件更新 $A^{-1}_{\Omega_1,\Omega_1}r_{\Omega_1}$ 对应于仅在 Ω_1 上求解泊松方程. 类似地，可利用依赖于 x_i 在节点5和6上的边界条件，更新 $A^{-1}_{\Omega_2,\Omega_2}r_{\Omega_2}$.

在 Ω_i 是长方形的情况下，前面的任何一种快速方法，如多重网格法可用于求解 $A^{-1}_{\Omega_i,\Omega_i}r_{\Omega_i}$. 因为加性施瓦茨方法是迭代的，所以在 Ω_i 上精确地求解问题是不必要的.

实际上，加性施瓦茨方法的特色是可用来作像共轭梯度法那样的克雷洛夫子空间方法的预条件子(见6.6.5节). 按6.6.5节的记号，预条件子 M 由下式给定

353

$$M^{-1}=\left[\begin{array}{c|c}A^{-1}_{\Omega_1,\Omega_1}&0\\\hline 0&0\end{array}\right]+\left[\begin{array}{c||c}0&0\\\hline\hline 0&A^{-1}_{\Omega_2,\Omega_2}\end{array}\right].$$

若 Ω_1 和 Ω_2 不交叠，则 M^{-1} 将简化为

$$\begin{bmatrix}A^{-1}_{\Omega_1,\Omega_1}&0\\0&A^{-1}_{\Omega_2,\Omega_2}\end{bmatrix}$$

并且我们将执行块雅可比迭代. 但是我们知道雅可比法收敛得不特别快，因为来自一个区域有关解的“信息”只能穿过它们之间的边界缓慢地移动到其他区域(见6.9节开始处的讨论). 但是只要交叠是两个区域的一个足够大的片断，信息将很快地传播，足以保证快速收敛. 当然不要太大的交叠，因为这会增加相当大的工作量. 设计一个好的区域分解方法的目标是：选择区域和交叠，只要做尽可能少的工作就有快速的收敛；下面更多地叙述收敛如何依赖于交叠.

由6.5节的讨论，我们知道高斯-塞德尔法似乎比雅可比法更有效. 在这里最好还是用交叠块高斯-塞德尔法(overlapping block Gauss-Seidel method)(更一般地称为乘性施瓦茨方法(multiplicative schwarz method))，它的速度通常是加性块雅可比迭代

法（加性施瓦茨方法）的两倍.

算法 6.19 更新 $Ax=b$ 的近似解 x_i 的乘性施瓦茨方法：

(1) $r_{\Omega_1}=(b-Ax_i)_{\Omega_1}$ /* 在 Ω_1 上计算 x_i 的残差 */

(2) $x_{i+\frac{1}{2},\Omega_1}=x_{i,\Omega_1}+A^{-1}_{\Omega_1,\Omega_1}\cdot r_{\Omega_1}$ /* 在 Ω_1 更新解 */

(2′) $x_{i+\frac{1}{2},\Omega\setminus\Omega_1}=x_{i,\Omega\setminus\Omega_1}$

(3) $r_{\Omega_2}=(b-Ax_{i+\frac{1}{2}})_{\Omega_2}$ /* 在 Ω_2 上计算 $x_{i+\frac{1}{2}}$ 的残差 */

(4) $x_{i+1,\Omega_2}=x_{i+\frac{1}{2},\Omega_2}+A^{-1}_{\Omega_2,\Omega_2}\cdot r_{\Omega_2}$ /* 在 Ω_2 上更新解 */

(4′) $x_{i+1,\Omega\setminus\Omega_2}=x_{i+\frac{1}{2},\Omega\setminus\Omega_2}$

注意，倘若 $x_{i+\frac{1}{2}}$ 和 x_{i+1} 覆盖 x_i 的话，那么第(2′)行和第(4′)行不需要移动任何数据.

这个算法象算法 6.18 一样，首先利用来自 x_i 的边界数据，在 Ω_1 上求解泊松方程. 然后在 Ω_2 上求解泊松方程，但是是利用刚才更新过的边界数据. 它也可用作为克雷洛夫子空间方法的一个预条件子.

实际上用于多于两个区域（Ω_1 和 Ω_2）的区域. 当解的区域更复杂或当有许多独立的并行处理机可以用于求解独立的问题 $A^{-1}_{\Omega_i,\Omega_i}r_{\Omega_i}$，或只保存小的子问题 $A^{-1}_{\Omega_i,\Omega_i}r_{\Omega_i}$ 并且求解代价不高时，才使用较多的区域.

下面是关于模型问题以及类似的椭圆型偏微分方程有关计算方法理论收敛性分析的小结. 设 h 是网孔间距. 收敛作为 h 的一个函数，当 h 减少到 0 时. 这一理论能够预估收敛需多少次迭代. 在两个区域的情况下，只要交叠区域 $\Omega_1\cap\Omega_2$ 是总区域 $\Omega_1\cup\Omega_2$ 的一个非零片断，当 h 趋于 0 时，收敛所需要的迭代次数与 h 无关. 这是一个具有吸引力的性质，让人回想到多重网格法，它也是以与网格大小 h 无关的速率收敛的. 但是一次迭代，包括在 Ω_1 和 Ω_2 精确地求解子问题的代价可能比原问题更高. 因而除非在 Ω_1 和 Ω_2 上的解非常便宜（正如上面的 L-形状区域那样），否则代价仍然是高的.

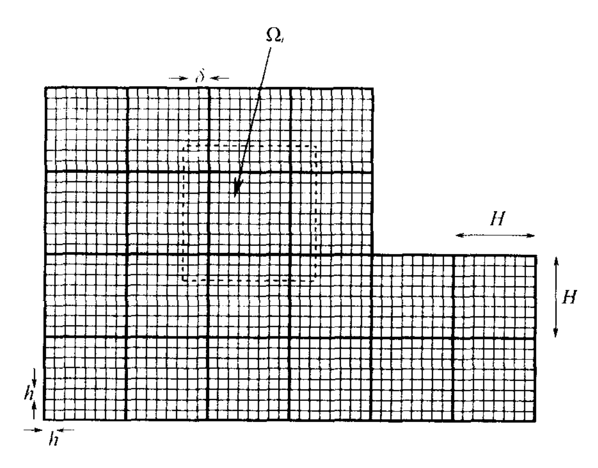
[354]

现在假定有许多区域 Ω_i，每个的尺寸 $H\gg h$. 换言之，把 Ω_i 看作具有间距 H 的粗网孔界定的区域，加上某些超出边界的单元，在图 6-21 中由虚线所示.

设 $\delta<H$ 是相邻的区域交叠的数量. 今设 H，δ 和 h 都趋于零，使得交叠分数 δ/H 保持常数且 $H\gg h$. 则收敛所需的迭代次数像 $1/H$ 那样增长，即与细网孔间距 h 无关. 这个结论接近于多重网格法，但仍不如多重网格法执行常数次迭代并且每个未知量做 $O(1)$ 次工作.

要达到多重网格法的性能，需要更多的想法，这一想法或许会与多重网格法相似，这一点也不奇怪. 使用间隔为 H 的粗网格上的问题的一个近似 A_H，得到一个粗网格预条件子（coarse grid preconditioner）加到细网格预条件子 $A^{-1}_{\Omega_i,\Omega_i}$ 之上. 这一算法需

要三个矩阵来描述. 首先，设 A_H 是用粗网孔间距 H 离散模型问题的矩阵. 其次，需要一个限制算子(restriction operator) R，在细网孔上取残差，并把它限制在粗网孔上的值；这本质上与多重网格法中相同(见6.9.2节). 最后，需要一个插值算子(interpolation operator)取粗网孔上的值并把它们插值到细网孔上；正如多重网格那样，这也出现 $\boldsymbol{R}^{\mathrm{T}}$.



[355]

图6-21 L-形状区域的粗、细离散化

算法6.20 更新 $\boldsymbol{Ax}=\boldsymbol{b}$ 的近似解 x_i 得到较好的解 x_{i+1} 的两层加性施瓦茨方法：

$x_{i+1}=x_i$

for $i=1$ to 区域 Ω_i 的编号

  $r_{\Omega_i}=(b-\boldsymbol{A}x_i)_{\Omega_i}$

  $x_{i+1,\Omega_i}=x_{i+1,\Omega_i}+\boldsymbol{A}_{\Omega_i,\Omega_i}^{-1}\cdot r_{\Omega_i}$

end for

$x_{i+1}=x_{i+1}+\boldsymbol{R}^{\mathrm{T}}\boldsymbol{A}_C^{-1}\boldsymbol{R}r$

正如算法6.18一样，这个方法的特色是可用来作为克雷洛夫子空间方法的预条件子.

这个算法的收敛性理论也可应用于比泊松方程更一般的问题，假定当 H、δ 和 h 减少到0同时 δ/H 保持固定时，收敛所需要的迭代次数与 H, h 或 δ 无关. 这意味着只要求解子问题 $\boldsymbol{A}_{\Omega_i,\Omega_i}^{-1}$ 和 $\boldsymbol{A}_H^{-1}$ 的工作与未知量成比例，则其复杂性就像多重网格一样好.

在现实世界问题中实施这些方法可能是复杂的，这对读者来说可能是显而易见的. 存在可利用的在线(on-line)软件来实施这里描述的许多模块，而且也能在并行机上运

行它们. 它被称为PETSc(科学计算的可移植易扩展的工具箱). 在http://www.mcs.anl.gov/petsc/petsc.html上可以得到PETSc并且在[232]中有它的简明描述.

6.11 第6章的参考书目和其他话题

现代迭代法的最新综述在[15,107,136,214]中给出，它们的并行实现也在[76]中加以综述. 经典的方法，诸如雅可比、高斯-塞德尔和SOR方法，也在[249,137]中详尽地加以讨论. 多重网格法在[43,185,186,260,268]及那里的参考文献中讨论；[91]是一个环球网站，提供了指向广泛的文献、软件等信息的链接. 区域分解在[49,116,205,232]中讨论. 切比雪夫及其他多项式在[240]中讨论. FFT在任何关于计算机科学算法的好教科书，例如[3]和[248]中讨论. 在[47,46]中建立块循环约化的一种稳定的形式.

6.12 第6章问题

356

问题6.1(容易) 证明引理6.1.

问题6.2(容易) 对$\boldsymbol{T}_N$的三角分解证明下列公式.

1. 楚列斯基分解$\boldsymbol{T}_N=\boldsymbol{B}_N^{\mathrm{T}}\boldsymbol{B}_N$有上双对角楚列斯基因子，其中

$$B_N(i,i)=\sqrt{\frac{i+1}{i}}\text{和}B_N(i,i+1)=\sqrt{\frac{i}{i+1}}.$$

2. 对$\boldsymbol{T}_N$用部分选主元的高斯消元法的结果是$\boldsymbol{T}_N=\boldsymbol{L}_N\boldsymbol{U}_N$，其中三角形因子是上双对角的

$$\boldsymbol{L}_N(i,i)=1\text{ 和 }\boldsymbol{L}_N(i+1,i)=-\frac{i}{i+1},$$

$$\boldsymbol{U}_N(i,i)=\frac{i+1}{i}\text{和 }\boldsymbol{U}_N(i,i+1)=-1.$$

3. $\boldsymbol{T}_N=\boldsymbol{D}_N\boldsymbol{D}_N^{\mathrm{T}}$，其中$\boldsymbol{D}_N$是在主对角线上为1和在上对角线上为$-1$的$N\times(N+1)$阶上双对角阵.

问题6.3(容易) 证实(6.13)式.

问题6.4(容易)

1. 证明引理6.2.
2. 证明引理6.3.
3. 证明西尔维斯特方程$\boldsymbol{AX}-\boldsymbol{XB}=\boldsymbol{C}$等价于$(\boldsymbol{I}_n\otimes\boldsymbol{A}-\boldsymbol{B}^{\mathrm{T}}\otimes\boldsymbol{I}_m)\mathrm{vec}(\boldsymbol{X})=\mathrm{vec}(\boldsymbol{C})$.
4. 证明$\mathrm{vec}(\boldsymbol{AXB})=(\boldsymbol{B}^{\mathrm{T}}\otimes\boldsymbol{A})\cdot\mathrm{vec}(\boldsymbol{X})$.

问题6.5(中等) 假定$\boldsymbol{A}^{n\times n}$可对角化，故$\boldsymbol{A}$有$n$个无关的特征向量$\boldsymbol{A}x_i=\alpha_i x_i$，或$\boldsymbol{AX}=\boldsymbol{X}\boldsymbol{\Lambda}_A$，其中$\boldsymbol{X}=[x_1,\cdots,x_n]$和$\boldsymbol{\Lambda}_A=\mathrm{diag}(\alpha_i)$. 类似地，假如$\boldsymbol{B}^{m\times m}$可对角化，

故 $\boldsymbol{B}$ 有 m 个无关的特征向量：$\boldsymbol{B}y_i=\beta_i y_i$，或 $\boldsymbol{BY}=\boldsymbol{Y}\boldsymbol{\Lambda}_B$，其中 $\boldsymbol{Y}=[y_1,\cdots,y_m]$和 $\boldsymbol{\Lambda}_B=\mathrm{diag}(\beta_j)$. 证明下列结论.

1. $\boldsymbol{I}_m\otimes\boldsymbol{A}+\boldsymbol{B}\otimes\boldsymbol{I}_n$ 的 mn 个特征值是 $\lambda_{ij}=\alpha_i+\beta_j$，即 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的特征值对应的一切可能的和. 对应的特征向量是 $\boldsymbol{z}_{ij}$，其中 $\boldsymbol{z}_{ij}=\boldsymbol{x}_i\otimes\boldsymbol{y}_i$，其第$(km+l)$个元素是 $\boldsymbol{x}_i(k)\boldsymbol{y}_i(l)$. 另一种写法为

$$(\boldsymbol{I}_m\otimes\boldsymbol{A}+\boldsymbol{B}\otimes\boldsymbol{I}_n)(\boldsymbol{Y}\otimes\boldsymbol{X})=(\boldsymbol{Y}\otimes\boldsymbol{X})\cdot(\boldsymbol{I}_m\otimes\boldsymbol{\Lambda}_A+\boldsymbol{\Lambda}_B\otimes\boldsymbol{I}_n).\tag{6.63}$$

2. 西尔维斯特方程 $\boldsymbol{AX}+\boldsymbol{XB}^{\mathrm{T}}=\boldsymbol{C}$ 非奇异(给定任意的 $\boldsymbol{C}$ 求解 $\boldsymbol{X}$)当且仅当对一切 $\boldsymbol{A}$ 的特征值 α_i 和 $\boldsymbol{B}$ 的特征值 β_j，和 $\alpha_i+\beta_j\neq0$. 对稍为不同的西尔维斯特方程 $\boldsymbol{AX}+\boldsymbol{XB}=\boldsymbol{C}$ 同样成立.（也可见问题4.6）

357

3. $\boldsymbol{A}\otimes\boldsymbol{B}$ 的 mn 个特征值是 $\lambda_{ij}=\alpha_i\beta_j$，即 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的特征值对的一切可能积. 对应的特征向量是 z_{ij}，其中 $z_{ij}=\boldsymbol{x}_i\otimes\boldsymbol{y}_j$，其第$(km+l)$个元素是 $\boldsymbol{x}_i(k)\boldsymbol{y}_j(l)$. 另一种写法为

$$(\boldsymbol{B}\otimes\boldsymbol{A})(\boldsymbol{Y}\otimes\boldsymbol{X})=(\boldsymbol{Y}\otimes\boldsymbol{X})\cdot(\boldsymbol{\Lambda}_B\otimes\boldsymbol{\Lambda}_A).\tag{6.64}$$

问题6.6(容易，程序设计)　编写执行算法6.2的单行 Matlab 程序：对泊松方程的单步雅可比算法. 对它进行测试，证实它的收敛速度像6.5.4节中所预估的一样快.

问题6.7(困难)　证明引理6.7.

问题6.8(中等、程序设计)　编写一个利用 FFT 在正方形上求解离散的模型问题的 Matlab 程序. 输入是维数 N 和值为 f_{ij} 的 $N\times N$ 阶矩阵. 输出是解 v_{ij} 的 $N\times N$ 阶矩阵和残差 $\|\boldsymbol{T}_{N\times N}\boldsymbol{v}-h^2f\|_2/(\|\boldsymbol{T}_{N\times N}\|_2\cdot\|\boldsymbol{v}\|)$. 你也应制作一个关于 f 和 $\boldsymbol{v}$ 的三维图. 利用 Matlab 中建立的 FFT. 如果利用你能够利用的一切 Matlab 特征，则你的程序一定不会超过几行长. 求解几个你知道解的问题和几个你不知道解的问题.

1. $f_{jk}=\sin(j\pi/(N+1))\cdot\sin(k\pi/(N+1))$.

2. $f_{jk}=\sin(j\pi/(N+1))\cdot\sin(k\pi/(N+1))+\sin(3j\pi/(N+1))\cdot\sin(5k\pi/(N+1))$.

3. f 有几个尖峰信号(同时为正和负)并且其余为0. 这个函数近似位于尖峰信号上的带电粒子的静电势并且具有和尖峰信号高度(正或负)成比例的电荷，若尖峰信号全正，则这也是重力势.

问题6.9(中等)　证实通过从右到左执行矩阵-向量乘法计算(6.47)中的公式数学上是和算法6.13相同的.

问题6.10(中等；困难)

1. (困难)　设 $\boldsymbol{A}$ 和 $\boldsymbol{H}$ 是可换的(commute)实对称 $n\times n$ 阶矩阵，即 $\boldsymbol{AH}=\boldsymbol{HA}$. 证明存在一个正交阵 $\boldsymbol{Q}$ 使 $\boldsymbol{QAQ}^{\mathrm{T}}=\mathrm{diag}(\alpha_1,\cdots,\alpha_n)$ 和 $\boldsymbol{QHQ}^{\mathrm{T}}=\mathrm{diag}(\theta_1,\cdots,\theta_n)$ 都是对角阵. 换言之，$\boldsymbol{A}$ 和 $\boldsymbol{H}$ 有相同的特征向量. 提示：首先假定 $\boldsymbol{A}$ 有不同的特征值，然后去掉这个假设.

358

2. (中等)设

$$\hat{T} = \begin{bmatrix} \alpha & \theta & & \\ \theta & \ddots & \ddots & \\ & \ddots & \ddots & \theta \\ & & \theta & \alpha \end{bmatrix}$$

是对称三对角特普利兹矩阵，即沿对角线有常数 α 和沿非对角线有常数 θ 的对称三对角矩阵. 写出 T 的特征值和特征向量的简单公式. 提示：利用引理6.1.

3.（困难）设

$$T = \begin{bmatrix} A & H & & \\ H & \ddots & \ddots & \\ & \ddots & \ddots & H \\ & & H & A \end{bmatrix}$$

是 $n^2 \times n^2$ 阶块三对角阵. 沿对角线有 n 个相同的 A，如上面一样设 $QAQ^T = \text{diag}(\alpha_1, \cdots, \alpha_n)$ 是 A 的特征分解并设 $QHQ^T = \text{diag}(\theta_1, \cdots, \theta_n)$ 是 H 的特征分解. 根据 α_i，θ_i 和 Q 写出 T 的 n^2 个特征值和特征向量的简单公式. 提示：利用克罗内克积.

4.（中等） 指出如何以 $O(n^3)$ 次运算求解 $Tx = b$. 与之对比，稠密LU分解和带状LU分解的运算次数多大?

5.（中等） 假如 A 和 H 是上面定义的（可能不同的）对称三对角特普利兹阵. 证明如何利用FFT只用 $O(n^2\log n)$ 次运算求解 $Tx = b$.

问题6.11（容易） 假如 R 是上三角非奇异阵而 C 是上海森伯格矩阵. 证明 RCR^{-1} 是上海森伯格矩阵.

问题6.12（中等） 证明克雷洛夫子空间 $K_k(A, y_1)$ 维数为 k 当且仅当阿诺尔迪算法（算法6.9）或兰乔斯算法（算法6.10）可计算 q_k 而不会先行停止.

问题6.13（中等）证明：当 $A^{n\times n}$ 对称正定和 $Q^{n\times k}$ 列满秩时，$T = Q^TAQ$ 也对称正定（对此问题，Q 不一定是正交的）.

问题6.14（中等） 证明定理6.9.

359

问题6.15（中等；困难）

1.（中等）证实(6.58)式.

2.（中等）证实(6.60)式.

3.（困难）证明定理6.11.

问题6.16（中等，程序设计） 运用多重网格法在正方形上求解离散的模型问题的Matlab程序在经典的网页HOMEPAGE/Matlab/MG_README. html上可以得到. 首先运行演示（执行"makemgdemo"然后执行"testfmgv"）. 然后尝试运行testfmgv对不同的右端项（输入数组 b），每次递归调用多重网格解法器前后不同的加权雅可比迭代次数（输入jac1和jac2），以及不同的迭代次数（输入iter）. 软件将画出收敛速率（连贯的残差之比）；这个比与 b 的大小有关吗? 与 b 中的频率有关吗? 与jac1和jac2中的值有关吗? jac1和jac2哪个值是最有效的解?

问题6.17（中等；程序设计）　利用来自问题6.8或者问题6.16的快速模型问题解算器，利用区域分解去构造6.10节中所描述的L-形状区域上的泊松方程快速解算器. 大的正方形是1×1而小的正方形是$.5\times.5$的，它们附在大的正方形的右下部分. 计算残差以证明你的解答是正确的.

问题6.18（困难）　填补像表6-1那样一个表的表列数据，但是是为了求解三维而不是二维泊松方程. 假定未知量的网格是$N\times N\times N$，$n=N^3$，试着尽你所能地填补第2列和第3列的表列数据.

360

第 7 章 特征值问题的迭代方法

7.1 概述

在本章中讨论求矩阵特征值的迭代法，这类矩阵阶数太大以致无法使用第 4 章和第 5 章的直接法. 换言之，我们寻找远小于 $O(n^2)$ 存储量和 $O(n^3)$ flops 的算法. 为了表示大多数 $n\times n$ 阶矩阵的特征向量，将采用 n^2 存储量，这意味着对一个矩阵而言，我们寻找的算法只计算几个用户选择的特征值和特征向量.

我们的讨论将依赖于 6.6 节中建立的克雷洛夫子空间方法的材料，5.2 节中关于对称特征值问题的材料，以及 5.3 节中关于幂法和逆迭代的材料. 建议读者复习这些章节.

最简单的特征值问题是仅仅计算绝对值最大的特征值连同它的特征向量. 幂法(算法 4.1)是适合这项任务的最简单的算法：记得它的内循环是

$$\boldsymbol{y}_{i+1}=\boldsymbol{A}\boldsymbol{x}_i,$$
$$\boldsymbol{x}_{i+1}=\boldsymbol{y}_{i+1}/\|\boldsymbol{y}_{i+1}\|_2,$$

其中 x_i 收敛到对应于所要求的特征向量(倘若只有一个绝对值最大的特征值，且 x_1 不位于包含它的特征向量的不变子空间中). 注意算法只使用 $\boldsymbol{A}$ 执行矩阵 - 向量乘法，故运行算法所需的全部工作是一个取 $\boldsymbol{x}_i$ 作为输入而返回 $\boldsymbol{A}\boldsymbol{x}_i$ 作为输出的“黑箱”(black-box)(见例 6.13).

一个密切相关的问题是求最接近于用户提供的值 σ 的特征值连同它的特征向量. 这正好是设计逆迭代(算法 4.2)去处理的情况. 记得它的内循环是

$$\boldsymbol{y}_{i+1}=(\boldsymbol{A}-\sigma\boldsymbol{I})^{-1}\boldsymbol{x}_i,$$
$$\boldsymbol{x}_{i+1}=\boldsymbol{y}_{i+1}/\|\boldsymbol{y}_{i+1}\|_2,$$

即求解具有系数矩阵为 $\boldsymbol{A}-\sigma\boldsymbol{I}$ 的线性方程组. 倘若只有一个最接近于 σ 的特征值，则 $\boldsymbol{x}_i$ 再次收敛到要求的特征向量(并且 $\boldsymbol{x}_1$ 满足如前面一样的性质). 第 6 章或 2.7.4 节中的任何稀疏矩阵技巧可用于求解 $\boldsymbol{y}_{i+1}$，尽管这些方法通常比简单地用 $\boldsymbol{A}$ 相乘的代价昂贵得多. 当 $\boldsymbol{A}$ 对称时，瑞利商迭代(算法 5.1)可用于加速收敛(虽然它不总是保证收敛于 A 的最接近于 σ 的特征值).

从给定的 $\boldsymbol{x}_1$ 出发，用幂法或者用逆迭代的 $k-1$ 次迭代产生一个向量序列 $\boldsymbol{x}_1$，$\boldsymbol{x}_2,\cdots\boldsymbol{x}_k$. 这些向量张成 6.6.1 节中定义的克雷洛夫子空间. 在幂法的情况下，这个克雷洛夫子空间是 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{x}_1)=[\boldsymbol{x}_1,\boldsymbol{A}\boldsymbol{x}_1,\boldsymbol{A}^2\boldsymbol{x}_1,\cdots,\boldsymbol{A}^{k-1}\boldsymbol{x}_1]$，而在逆迭代的情况下，这

个克雷洛夫子空间是 $\mathcal{K}_k((A-\sigma I)^{-1},x_1)$. 与其取 x_k 作为近似的特征向量，倒不如自然地去 $\mathcal{K}_k$ 中求“最佳”逼近的特征向量；即最佳的线性组合 $\sum_{i=1}^{k}\alpha_i x_i$. 采用与6.6.2节中求解 $Ax=b$ 同样的方法，从 $\mathcal{K}_k$ 中寻找 $Ax=b$ 的最佳逼近解. 我们将看到来自 $\mathcal{K}_k$ 的最佳逼近特征向量(和特征值)比单独的 x_k 好很多. 因为(一般而言) $\mathcal{K}_k$ 的维数为 k，所以实际上可利用它计算 k 个最佳逼近特征值和特征向量，这些最佳的近似称为里茨值(Ritz value)和里茨向量(Ritz vector).

我们将集中讨论对称情况 $A=A^{\mathrm{T}}$. 最后一节中将简要地描述非对称情况.

本章的其余部分组织如下：7.2节讨论瑞利-里茨方法，这是从克雷洛夫子空间中获取有关特征值和特征向量的基本方法. 7.3节以精确算术运算讨论主要算法——兰乔斯算法. 7.4节以浮点算术运算分析兰乔斯算法的相当不同的性态，而7.5和7.6节不顾舍入描述计算可靠解答的兰乔斯算法的实际执行过程. 最后，7.7节简要地讨论非对称特征问题的算法.

7.2　瑞利-里茨方法

设 $Q=[Q_k,Q_u]$ 是任意的 $n\times n$ 阶正交阵，其中 Q_k 是 $n\times k$ 阶而 Q_u 是阶 $n\times(n-k)$ 阶矩阵. 实际上 Q_k 的列将由兰乔斯算法(算法6.10或下页的算法7.1)算出并张成克雷洛夫子空间 $\mathcal{K}_k$，下标 u 指示 Q_u(通常)是未知的. 从现在起不关心从何处得到 Q.

我们将使用下列记号(在(6.31)式中也使用过)：

362

$$T=Q^{\mathrm{T}}AQ=[Q_k,\ Q_u]^{\mathrm{T}}A[Q_k,\ Q_u]=\begin{bmatrix}Q_k^{\mathrm{T}}AQ_k & Q_k^{\mathrm{T}}AQ_u\\ Q_u^{\mathrm{T}}AQ_k & Q_u^{\mathrm{T}}AQ_u\end{bmatrix}$$

$$\equiv \begin{matrix} \\ k\\ n-k\end{matrix}\ \begin{matrix}\begin{matrix}k & n-k\end{matrix}\\ \begin{bmatrix}T_k & T_{uk}\\ T_{ku} & T_u\end{bmatrix}\end{matrix}=\begin{bmatrix}T_k & T_{ku}^{\mathrm{T}}\\ T_{ku} & T_u\end{bmatrix}. \tag{7.1}$$

当 $k=1$ 时，T_k 正好是瑞利商 $T_1=\rho(Q_1,A)$(见定义5.1). 因而对 $k>1$，T_k 是瑞利商的一个自然的推广.

定义7.1　瑞利-里茨过程利用 $T_k=Q_k^{\mathrm{T}}AQ_k$ 的特征值近似 A 的特征值. 这些近似称为里茨值. 设 $T_k=V\Lambda V^{\mathrm{T}}$ 是 T_k 的特征分解. 对应的特征向量近似是 Q_kV 的列并且称为里茨向量.

里茨值和里茨向量被认为是 A 的特征值和特征向量的最佳近似有几个原因. 首先，当 Q_k 因而 T_k 已知，但 Q_u 未知因而 T_{ku} 和 T_u 未知时，里茨值和里茨向量是来自矩阵已知部分的自然近似. 其次，它们满足下列定理5.5的推广.（定理5.5证明瑞利商是单个特征值的一个“最佳近似”）. 记得 Q_k 的列张成 A 的不变子空间当且仅当对某个矩阵 R，$AQ_k=Q_kR$.

定理7.1 取遍所有$k\times k$阶对称阵$\boldsymbol{R}$，$\|\boldsymbol{A}\boldsymbol{Q}_k-\boldsymbol{Q}_k\boldsymbol{R}\|_2$的极小值在$\boldsymbol{R}=\boldsymbol{T}_k$时达到，此时，$\min\|\boldsymbol{A}\boldsymbol{Q}_k-\boldsymbol{Q}_k\boldsymbol{R}\|_2=\|\boldsymbol{T}_{ku}\|_2$. 设$\boldsymbol{T}_k=\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$是$\boldsymbol{T}_k$的特征分解. 取遍所有$n\times k$阶正交阵$\boldsymbol{P}_k$，其中span（$\boldsymbol{P}_k$）=span（$\boldsymbol{Q}_k$）并且取遍所有对角阵$\boldsymbol{D}$，$\|\boldsymbol{A}\boldsymbol{P}_k-\boldsymbol{P}_k\boldsymbol{D}\|_2$的极小值也是$\|\boldsymbol{T}_{ku}\|_2$而且在$\boldsymbol{P}_k=\boldsymbol{Q}_k\boldsymbol{V}$和$\boldsymbol{D}=\boldsymbol{\Lambda}$时达到.

换言之，在残差$\|\boldsymbol{A}\boldsymbol{P}_k-\boldsymbol{P}_k\boldsymbol{D}\|_2$极小的意义下，$\boldsymbol{Q}_k\boldsymbol{V}$的列(里茨向量)是“最佳的”近似特征向量并且$\boldsymbol{\Lambda}$的对角元(里茨值)是“最佳的”近似特征值.

证明 为简化记号临时去掉$\boldsymbol{T}_k$和$\boldsymbol{Q}_k$的下标，故可记$k\times k$阶矩阵$\boldsymbol{T}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$. 设$\boldsymbol{R}=\boldsymbol{T}+\boldsymbol{Z}$. 我们应该证明当$\boldsymbol{Z}=0$时$\|\boldsymbol{A}\boldsymbol{Q}-\boldsymbol{Q}\boldsymbol{R}\|_2^2$达到极小. 利用毕达哥拉斯定理的变相形式来做:

$$
\begin{aligned}
\|AQ-QR\|_2^2 &= \lambda_{\max}[(AQ-QR)^{\mathrm{T}}(AQ-QR)] && \text{由引理 1.7 的第 7 部分}\\
&= \lambda_{\max}[(AQ-Q(T+Z))^{\mathrm{T}}(AQ-Q(T+Z))]\\
&= \lambda_{\max}[(AQ-QT)^{\mathrm{T}}(AQ-QT)-(AQ-QT)^{\mathrm{T}}(QZ)\\
&\quad -(QZ)^{\mathrm{T}}(AQ-QT)+(QZ)^{\mathrm{T}}(QZ)]\\
&= \lambda_{\max}[(AQ-QT)^{\mathrm{T}}(AQ-QT)-(Q^{\mathrm{T}}AQ-T)Z-Z^{\mathrm{T}}(Q^{\mathrm{T}}AQ-T)+Z^{\mathrm{T}}Z]\\
&= \lambda_{\max}[(AQ-QT)^{\mathrm{T}}(AQ-QT)+Z^{\mathrm{T}}Z] && \text{因为 } Q^{\mathrm{T}}AQ=T\\
&\geqslant \lambda_{\max}[(AQ-QT)^{\mathrm{T}}(AQ-QT)] && \text{由问题 5.5, 因为 } Z^{\mathrm{T}}Z \text{ 是对称半正定的.}\\
&= \|AQ-QT\|_2^2 && \text{由引理 1.7 的第 7 部分}
\end{aligned}
$$

363

恢复下标，容易计算极小值.

$$\|\boldsymbol{A}\boldsymbol{Q}_k-\boldsymbol{Q}_k\boldsymbol{T}_k\|_2=\|(\boldsymbol{Q}_k\boldsymbol{T}_k+\boldsymbol{Q}_u\boldsymbol{T}_{ku})-(\boldsymbol{Q}_k\boldsymbol{T}_k)\|_2=\|\boldsymbol{Q}_u\boldsymbol{T}_{ku}\|_2=\|\boldsymbol{T}_{ku}\|_2.$$

若用任意的积$\boldsymbol{Q}_k\boldsymbol{U}$代替$\boldsymbol{Q}_k$，其中$\boldsymbol{U}$是另一个正交阵，则$\boldsymbol{Q}_k$和$\boldsymbol{Q}_k\boldsymbol{U}$的列张成相同的空间，并且

$$\|\boldsymbol{A}\boldsymbol{Q}_k-\boldsymbol{Q}_k\boldsymbol{R}\|_2=\|\boldsymbol{A}\boldsymbol{Q}_k\boldsymbol{U}-\boldsymbol{Q}_k\boldsymbol{R}\boldsymbol{U}\|_2=\|\boldsymbol{A}(\boldsymbol{Q}_k\boldsymbol{U})-(\boldsymbol{Q}_k\boldsymbol{U})(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{U})\|_2.$$

当$\boldsymbol{R}=\boldsymbol{T}_k$时，这些量仍然达到极小，通过选择$\boldsymbol{U}=\boldsymbol{V}$使$\boldsymbol{U}^{\mathrm{T}}\boldsymbol{T}_k\boldsymbol{U}$是对角阵，我们解决定理叙述中的第二个极小化问题. □

这个定理证实利用里茨值作为特征值近似是有道理的. 当利用兰乔斯算法计算$\boldsymbol{Q}_k$时，此时(见(6.31)式).

$$
\boldsymbol{T}=\left[\begin{array}{c|c}\boldsymbol{T}_k & \boldsymbol{T}_{ku}^{\mathrm{T}}\\ \hline \boldsymbol{T}_{ku} & \boldsymbol{T}_u\end{array}\right]=
\left[\begin{array}{cccc|cccc}
\alpha_1 & \beta_1 & & & & & & \\
\beta_1 & \ddots & \ddots & & & & & \\
 & \ddots & \ddots & \beta_{k-1} & & & & \\
 & & \beta_{k-1} & \alpha_k & \beta_k & & & \\
\hline
 & & & \beta_k & \alpha_{k+1} & \beta_{k+1} & & \\
 & & & & \beta_{k+1} & \ddots & \ddots & \\
 & & & & & \ddots & \ddots & \beta_{n-1}\\
 & & & & & & \beta_{n-1} & \alpha_n
\end{array}\right],
$$

则容易计算定理7.1中所有的量. 这是因为存在求对称三对角矩阵 $\boldsymbol{T}_k$ 的好的算法(见5.3节), 并且因为残差范数只不过是 $\|\boldsymbol{T}_{ku}\|_2=\beta_k$. (由兰乔斯算法知道 β_k 非负.)这就简化了下列定理中近似特征值和特征向量的误差界.

定理7.2 设 $\boldsymbol{T}_k$, $\boldsymbol{T}_{ku}$和 $\boldsymbol{Q}_k$ 如(7.1)式中那样定义. 设 $\boldsymbol{T}_k=\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$ 是 $\boldsymbol{T}_k$ 的特征分解, 364 其中 $\boldsymbol{V}=[\boldsymbol{v}_1,\cdots,\boldsymbol{v}_k]$是正交的而 $\boldsymbol{\Lambda}=\mathrm{diag}(\theta_1,\cdots,\theta_k)$. 则

1. 存在 $\boldsymbol{A}$ 的 k 个特征值 $\alpha_1,\cdots,\alpha_k$(不一定是最大的 k 个)使得对 $i=1,\cdots,k$. $|\theta_i-\alpha_i|\leqslant\|\boldsymbol{T}_{ku}\|_2$. 若 $\boldsymbol{Q}_k$ 用兰乔斯算法计算, 则 $|\theta_i-\alpha_i|\leqslant\|\boldsymbol{T}_{ku}\|_2=\beta_k$, 其中 β_k 是 $\boldsymbol{T}_{ku}$的右上角中的单个(也许)非零元.

2. $\|\boldsymbol{A}(\boldsymbol{Q}_k\boldsymbol{v}_i)-(\boldsymbol{Q}_k\boldsymbol{v}_i)\theta_i\|_2=\|\boldsymbol{T}_{ku}\boldsymbol{v}_i\|_2$. 于是, 里茨值 θ_i 和 $\boldsymbol{A}$ 的某个特征值 α 之间的差至多是 $\|\boldsymbol{T}_{ku}\boldsymbol{v}_i\|_2$, 它可能比 $\|\boldsymbol{T}_{ku}\|_2$ 小得多. 若 $\boldsymbol{Q}_k$ 由兰乔斯算法计算, 则 $\|\boldsymbol{T}_{ku}\boldsymbol{v}_i\|_2=\beta_k|\boldsymbol{v}_i(k)|$, 其中 $\boldsymbol{v}_i(k)$是 $\boldsymbol{v}_i$ 第 k 个(底部的)元素, 这个公式使我们很容易计算残差 $\|\boldsymbol{A}(\boldsymbol{Q}_k\boldsymbol{v}_i)-(\boldsymbol{Q}_k\boldsymbol{v}_i)\theta_i\|_2$, 即不用 $\boldsymbol{Q}_k$ 或用 $\boldsymbol{A}$ 乘任何向量.

3. 要是没有任何关于 $\boldsymbol{T}_u$ 的谱的进一步信息, 我们不能够推出任何基于里茨向量 $\boldsymbol{Q}_k\boldsymbol{v}_i$ 的有用的误差界. 若知道 θ_i 和 $\boldsymbol{T}_k$ 或 $\boldsymbol{T}_u$ 的其他任何特征值之间的间隙至少是 g, 则下式可以界定 $\boldsymbol{Q}_k\boldsymbol{v}_i$ 和 $\boldsymbol{A}$ 的一个正确的特征向量之间的夹角 θ,

$$\frac{1}{2}\sin 2\theta\leqslant\frac{\|\boldsymbol{T}_{ku}\|_2}{g}. \tag{7.2}$$

若由兰乔斯算法计算 $\boldsymbol{Q}_k$, 则界简化为

$$\frac{1}{2}\sin 2\theta\leqslant\frac{\beta_k}{g}.$$

证明

(1) $\hat{\boldsymbol{T}}=\begin{bmatrix}\boldsymbol{T}_k & 0\\ 0 & \boldsymbol{T}_u\end{bmatrix}$的特征值包括 θ_1 到 θ_k. 因为

$$\|\hat{\boldsymbol{T}}-\boldsymbol{T}\|_2=\left\|\begin{bmatrix}0 & \boldsymbol{T}_{ku}^{\mathrm{T}}\\ \boldsymbol{T}_{ku} & 0\end{bmatrix}\right\|_2=\|\boldsymbol{T}_{ku}\|_2,$$

外尔定理(定理5.1)告诉我们 $\hat{\boldsymbol{T}}$ 和 $\boldsymbol{T}$ 的特征值相差至多为 $\|\boldsymbol{T}_{ku}\|_2$. 但 $\boldsymbol{T}$ 和 $\boldsymbol{A}$ 的特征值是完全相同的, 所以证明了结论.

(2) 计算

$$\begin{aligned}\|\boldsymbol{A}(\boldsymbol{Q}_k\boldsymbol{v}_i)-(\boldsymbol{Q}_k\boldsymbol{v}_i)\theta_i\|_2 &= \|\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{Q}_k\boldsymbol{v}_i)-\boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{Q}_k\boldsymbol{v}_i)\theta_i\|_2\\ &= \left\|\begin{bmatrix}\boldsymbol{T}_k\boldsymbol{v}_i\\ \boldsymbol{T}_{ku}\boldsymbol{v}_i\end{bmatrix}-\begin{bmatrix}\boldsymbol{v}_i\theta_i\\ 0\end{bmatrix}\right\|_2=\left\|\begin{bmatrix}0\\ \boldsymbol{T}_{ku}\boldsymbol{v}_i\end{bmatrix}\right\|_2 \quad \text{因为 } \boldsymbol{T}_k\boldsymbol{v}_i=\theta_i\boldsymbol{v}_i\\ &= \|\boldsymbol{T}_{ku}\boldsymbol{v}_i\|_2.\end{aligned}$$

于是由定理5.5, $\boldsymbol{A}$ 有某个满足 $|\alpha-\theta_i|\leqslant\|\boldsymbol{T}_{ku}\boldsymbol{v}_i\|_2$ 的特征值 α. 若 $\boldsymbol{Q}_k$ 由兰乔斯算法 365 计算, 则 $\|\boldsymbol{T}_{ku}\boldsymbol{v}_i\|_2=\beta_k|\boldsymbol{v}_i(k)|$, 因为只有 $\boldsymbol{T}_{ku}$右上角的元素 β_k 非零.

(3) 再次使用例5.4说明要是没有关于 T_u 的谱的进一步信息，我们不能够推出关于里茨向量的有用的误差界：

$$T=\begin{bmatrix}1+g & \epsilon\\ \epsilon & 1\end{bmatrix},$$

其中 $0<\epsilon<g$. 设 $k=1$ 和 $Q_1=[e_1]$，因而 $T_1=[1+g]$ 而且近似特征向量就是 e_1. 但正如例5.4中所示 T 的特征向量接近于 $[1,\epsilon/g]^{\mathrm{T}}$ 和 $[-\epsilon/g,1]^{\mathrm{T}}$. 故没有 g 的下界，即 T_k 的特征值和所有其他的特征值，包括 T_u 的那些特征值之间的间隙不能界定计算的特征向量中的误差. 若我们有这样的一个下界，则可对 T 和 $T+E=\mathrm{diag}(T_k,T_u)$ 应用定理5.4的第二个界推出(7.2)式. □

7.3 精确算术运算的兰乔斯算法

求对称阵 A 的特征值之兰乔斯算法把构造克雷洛夫子空间的兰乔斯算法(算法6.10)与上节的瑞利-里茨过程结合起来. 换言之，与(7.1)式中一样，它构造正交兰乔斯向量的正交阵 $Q_k=[q_1,\cdots,q_k]$ 和用里茨值(对称三对角阵 $T_k=Q_k^{\mathrm{T}}AQ_k$ 的特征值)近似 A 的特征值.

算法7.1 求 $A=A^{\mathrm{T}}$ 的特征值和特征向量的精确算术运算兰乔斯算法

$q_1=b/\|b\|_2$, $\beta_0=0$, $q_0=0$
for $j=1$ to k
　　$z=Aq_j$
　　$\alpha_j=q_j^{\mathrm{T}}z$
　　$z=z-\alpha_j q_j-\beta_{j-1}q_{j-1}$
　　$\beta_j=\|z\|_2$
　　if $\beta_j=0$, quit
　　$q_{j+1}=z/\beta_j$
　　计算 T_j 的特征值，特征向量和误差界
end for

在本节中将通过稍稍详尽地描述一个数值例子来考察兰乔斯算法的收敛性. 选择的这个例子同时说明典型的收敛性态，也说明一些更多疑难的性态，这种性态称为误收敛(misconvergence). 因为初始向量 q_1 几乎正交于要求的特征值的特征向量或者当存在重(或非常接近的)特征值时可能发生误收敛.

366

本节的标题指出我们(几乎)排除本例中舍入误差的影响. 当然，Matlab代码(HOMEPAGE/Matlab/LanczosFullReorthog.m)用于产生下面按浮点算术运算运行的

例子，但是为了使它尽可能真实地模拟精确的结果，我们按一种特别精细且代价昂贵的方式执行兰乔斯算法(特别是算法 7.1 的内循环)．这个精细的执行过程称为完全再正交化的兰乔斯算法(Lanczos with full reorthogonalization)，正如下图的标题中指出的那样.

在下节中，将利用原来的算法 7.1 的廉价的执行过程来考察同样的数值例子，这个过程称为非再正交化的兰乔斯算法(Lanczos with no reorthogonalization)，目的是为了把它与完全再正交化的兰乔斯算法对比．(也将说明两个执行过程中的差别)．我们将看到原来的兰乔斯算法可能与更昂贵的"精确"算法有相当不同的表现．尽管如此，我们将指出如何利用不太昂贵的算法去可靠地计算特征值.

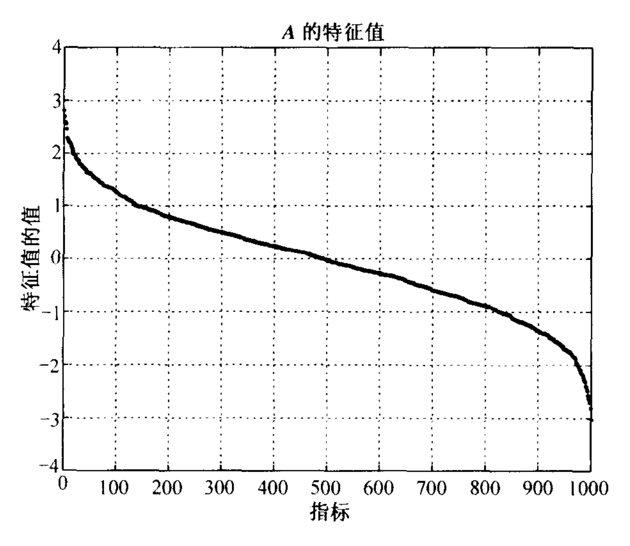


图 7-1　对角阵 A 的特征值

例 7.1　通过运行一个大的例子，1000 × 1000 阶对角阵 A，其特征值大部分随机地选自正态的高斯分布，来说明兰乔斯算法及其误差界．图 7-1 是一幅特征值的图．为使随后的图容易理解，我们也把 A 的对角元从最大的到最小的作了分类，故 $\lambda_i(A)=a_{ii}$ 具有对应的特征向量 e_i 为单位阵的第 i 列．存在几个极端的特征值并且其余的串靠近谱的中心．如下面所描述的那样初始兰乔斯向量 q_1，除了一个分量外所有分量都相等.

不失一般性在实验中用一个对角阵，因为用初始向量 q_1 对 A 运行兰乔斯算法等价于用初始向量 $Q^{\mathrm{T}}q_1$ 对 $Q^{\mathrm{T}}AQ$ 运行兰乔斯算法(见问题 7.1).

为了说明收敛性，将利用图 7-2(见彩插)中所示几种分类图．在图 7-2 中，上图第 k 列($k=1$ 到 9)和下图第 k 列($k=1$ 到 29)画出的表示每个 T_k 的特征值，在右面额外的列中画出 A 的特征值．因而第 k 列有 k 个加号，一个加号标记一个 T_k 的特征值．还用下列色彩 - 代码(color-coded)的特征值：每个 T_k 的最大和最小特征值用黑色表示，次大和次小特征值用红色表示，第三大和第三小特征值用绿色表示，而第四大

和第四小特征值用蓝色表示. 然后，这些色彩重复应用到谱的内部.

为了理解收敛性，考察每个 T_k 的最大特征值；这些黑色的加号都在每列的上部. 注意，它们随着 k 增大而单调递增，这是一个柯西交错定理的结果，因为 T_k 是 T_{k+1} 的一个子阵(见问题5.4). 事实上，柯西交错定理告诉我们更多的结论，T_k 的特征值与 T_{k+1} 的那些特征值交错(interlace)，即 $\lambda_i(T_{k+1}) \geq \lambda_i(T_k) \geq \lambda_{i+1}(T_{k+1}) \geq \lambda_{i+1}(T_k)$. 换言之，不只是 $i=1$(最大的特征值)，对任何固定的 i，$\lambda_i(T_k)$ 关于 k 单调递增. 这说明彩色的加号序列移动到图中的右上方.

对最小的特征值出现完全类似的现象：图7-2的每列底部的黑色加号表示每个 T_k 的最小的特征值，并且它们随着 k 增大而单调递减. 类似的，第 i 个最小的特征值也是单调地递减. 这也是柯西交错定理的一个简单的结论.

现在可以问随着 k 递增特征值 $\lambda_i(T_k)$ 可收敛于 A 的哪个特征值. 显然，T_k 的最大特征值 $\lambda_1(T_k)$ 应该收敛于 A 的最大特征值 $\lambda_1(A)$. 确实，若兰乔斯算法进行到 $k=n$ 步(不是因为某个 $\beta_k=0$ 过早地退出)，则 T_n 和 A 相似，故 $\lambda_1(T_n)=\lambda_1(A)$. 类似地，$T_k$ 的第 i 个最大的特征值 $\lambda_i(T_k)$ 必定单调递增并且收敛于 A 的第 i 个最大的特征值 $\lambda_i(A)$(倘若兰乔斯算法不过早地退出)，而且 T_k 的第 i 个最小的特征值 $\lambda_{k+1-i}(T_k)$ 必定类似地单调递减和收敛于 A 的第 i 个最小的特征值 $\lambda_{n+1-i}(A)$.

所有这些收敛的序列在图7-2中以及本节其他图中被表成共同颜色的加号序列. 考虑图7-2中下面的图形：对于大于15左右的 k，最上部和最下部的黑色加号构成水平行与最右边的列中画出的 A 的极端特征值相连；这就证明了收敛. 类似的，上面的红色加号序列构成水平行与最右边列中 A 的第二大特征值相连；它们比最外面的特征值稍后收敛. 图7-3(见彩插)的上面两个图像对更多兰乔斯算法步表示这个性态的放大情况. 367~368

总结上面讨论，极端特征值，即最大的和最小的特征值首先收敛，而内部的特征值最后收敛. 并且收敛是单调的，倘若兰乔斯算法不对某个 $\beta_k=0$ 过早地停止的话，T_k 的第 i 个最大的(最小的)特征值递增地(递减地)收敛于 A 的第 i 最大的(最小的)特征值.

现在更详尽地考察收敛性态，计算里茨值中实际的误差，并把这些误差同定理7.2的第2部分中的误差界作比较. 对图7-2中画出的相同的矩阵运行兰乔斯算法99步，并在图7-3中显示结果. 图7-3中左上图只表示最大的特征值，而右上图只表示最小的特征值. 369

图7-3中间的两个图像表示四个最大的计算特征值(在左边)和四个最小的计算特征值(在右边)中的误差. 在中间的图像中的颜色与上面图像中的颜色相匹配.

我们以三种方式度量和画出误差：

- 整体误差(global error)(实线)由 $|\lambda_i(T_k)-\lambda_i(A)|/|\lambda_i(A)|$ 给出. 用 $|\lambda_i(A)|$ 相除是为了规范化所有的误并位于1(不精确)和 10^{-16}(机器精度,或全精度)之间. 整体误差随 k 增大单调递减，除非兰乔斯算法过早地退出，我

们预期它减少到机器精度.

- *局部误差*(local error)(点线)由 $\min_j|\lambda_i(\boldsymbol{T}_k)-\lambda_j(\boldsymbol{A})|/|\lambda_i(\boldsymbol{A})|$给出. 局部误差度量 $\lambda_i(\boldsymbol{T}_k)$和 $\boldsymbol{A}$ 的最近的特征值 $\lambda_j(\boldsymbol{A})$，不只是最终的特征值 $\lambda_i(\boldsymbol{A})$之间的最小距离. 我们画出这个误差是因为有时局部误差比整体误差小得多.
- *误差界*(虚线)是由算法计算的量 $|\beta_k v_i(k)|/|\lambda_i(\boldsymbol{A})|$(只是用 $|\lambda_i(\boldsymbol{A})|$规范化，当然算法不知道 $|\lambda_i(\boldsymbol{A})|$!)

在图7-3中底部的两个图像表示兰乔斯向量 $\boldsymbol{q}_k$ 的特征向量分量，对应于四个最大的特征值的四个特征向量(在左边)和对应于四个最小的特征值的四个特征向量(在右边). 换言之，它们画 $\boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{e}_j=\boldsymbol{q}_k(j)$，其中 $\boldsymbol{e}_j$ 是对角阵 $\boldsymbol{A}$ 的第 j 个特征向量，对 $k=1$ 到99并且关于 $j=1$ 到4(在左边)和关于 $j=997$ 到1000(在右边). 分量根据对数尺度画出，用"+"和"○"分别指出分量是正还是负. 我们利用这些图帮助说明下面的收敛性.

现在利用图7-3更详尽地考察收敛性. $\boldsymbol{T}_k$ 的最大的特征值(图7-3左上图中最上面黑色的加号)开始马上收敛于它的最终值(大约2.81)，经兰乔斯25步之后精确到6位小数，50步之后精确到机器精度. 整体误差用中左图中的黑实线表示. 局部误差(点黑线)在不太多步之后与整体误差相同，虽然当一个特征值 $\lambda_i(\boldsymbol{T}_k)$ 在它到 $\lambda_i(\boldsymbol{A})$的途中碰巧下降接近于某个其他的 $\lambda_j(\boldsymbol{A})$时局部误差可能"意外地"小得多. 在同样的图中的黑虚线是由算法计算的相对误差界，直到75步左右它过高估计真正的误差. 相对误差界还正确地指出最大的特征值正确到几位小数.

370

第二大到第四大的特征值(在图7-3左上方图像中最上面的红色、绿色和蓝色加号)以类似的方式收敛，第 i 个特征值收敛稍稍快于第 $i+1$ 个特征值. 这是兰乔斯算法的典型性态.

图7-3的左下方图像根据特征向量分量 $\boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{e}_j$ 度量收敛性. 为说明这个图像，考虑当第一个特征值收敛时兰乔斯向量 $\boldsymbol{q}_k$ 发生什么情况. 收敛意味着对应的特征向量 $\boldsymbol{e}_1$ 几乎位于由兰乔斯向量张成的克雷洛夫子空间中. 特别因为第一个特征值经兰乔斯 $k=50$步之后收敛，这意味着 $\boldsymbol{e}_1$ 几乎必定是 $\boldsymbol{q}_1$ 到 $\boldsymbol{q}_{50}$的一个线性组合. 因为 $\boldsymbol{q}_k$ 相互正交，这意味着对 $k>50$，$\boldsymbol{q}_k$ 也必定正交于 $\boldsymbol{e}_1$. 这由左下方图像中的黑色曲线证实，通过50步它减小至小于 10^{-7}. 红色的曲线是 $\boldsymbol{q}_k$ 中 $\boldsymbol{e}_2$ 的分量，通过60步这个分量达到 10^{-8}. 绿色的曲线(第三个特征分量)和蓝色的曲线(第四个特征分量)通过若干步之后类似地变小.

现在讨论最小的四个特征值，通过图7-3右边的三个图像描述它们的性态. 我们选择了矩阵 $\boldsymbol{A}$ 和初始向量 $\boldsymbol{q}_1$，为了说明兰乔斯算法收敛性中可能产生的某些困难，而这种收敛性显示收敛不总是如刚才考察的四个特征值的情况那样简单.

特别，我们选择了 $\boldsymbol{q}_1(999)$，$\boldsymbol{q}_1$ 在第二个最小的特征值(-2.81)的方向的特征分量约为 10^{-7}，它比 $\boldsymbol{q}_1$ 的其他所有的分量(它们都相等)小 10^5 倍. 还有，我们选择的第三和第四个最小的特征值(编号为998和997)几乎相等：$-2.700\,001$ 和 -2.7.

类似于最大的特征值，T_k 的最小特征值收敛于 $\lambda_{1000}(A) \approx -3.03$ 是平凡的. 通过40步它精确到16位.

以红色表示的 T_k 的第二个最小的特征值在 -2.7 近旁，开始误收敛(misconverging)于 A 的第三个特征值. 确实，在图7-3的中右方图像中的红点线表明兰乔斯步 $40<k<50$ 时 $\lambda_{999}(T_k)$ 和 $\lambda_{998}(A)$ 有六位小数位相同. 相应的误差界(红色虚线)告诉我们对同样的 k，$\lambda_{999}(T_k)$ 与 A 的某个特征值有三到四位小数位相同. $\lambda_{999}(T_k)$ 误收敛的原因是克雷洛夫子空间开始具有一个对应的克雷洛夫子空间 e_{999} 的一个非常小的分量 10^{-7}. 这通过右下图中的红色曲线可以看出，它在 10^{-7} 上出发并在 e_{999} 的一个大的分量出现之前的45步一直取 10^{-7}. 当克雷洛夫子空间包含特征向量 e_{999} 的一个充分大的分量时，仅仅在这个时刻，$\lambda_{999}(T_k)$ 能重新开始收敛到它们最终值 $\lambda_{999}(A) \approx -2.81$，如右上图和中右图所示. 一旦这个收敛重新开始，$e_{999}$ 的分量开始重新减小并且当 $\lambda_{999}(T_k)$ 充分准确地收敛于 $\lambda_{999}(A)$ 时，它就变得非常小. (关于收敛率和特征分量 $q_1^T e_{999}$ 之间的数量关系，见下面讨论的 Kaniel 和 Saad 定理.)

371~372

事实上，若 q_1 精确地正交于 e_{999}，故 $q_1^T e_{999}=0$ 而不是 $q_1^T e_{999}=10^{-7}$，于是所有随后的兰乔斯向量也将正交于 e_{999}. 这意味着 $\lambda_{999}(T_k)$ 决不会收敛于 $\lambda_{999}(A)$. (证明见问题7.3). 在图7-4(见彩插)中说明这些，其中我们已稍稍修改 q_1 使得 $q_1^T e_{999}=0$. 注意无论何时都不出现 $\lambda_{999}(A) \approx -2.81$ 的近似.

幸亏当我们随机地选择 q_1 时，它正交于一个特征向量是极其靠不住的. 倘若我们没有错过任何特征值提供的"统计"迹象，我们总能以一个不同的随机的 q_1 重新运行兰乔斯算法.

"误收敛"另一个来源是(接近)重特征值，例如，第三个最小的特征值 $\lambda_{998}(A) = -2.700\,001$ 和第四个最小的特征值 $\lambda_{997} = -2.7$. 考虑 $\lambda_{998}(T_k)$，在图7-3的右上方和中间右边图像中最下面的绿色曲线，我们看出在兰乔斯步 $50<k<75$ 之间，$\lambda_{998}(T_k)$ 误收敛于 A 的两个最接近的特征值之间中途的(halfway) $-2.700\,000\,5$. 在由右上方图像提供的解答处这不是明显的，而在兰乔斯步数 $50<k<75$ 之间根据中间右边图像中绿色实线的水平线段这是明显的. 在76步再开始迅速地收敛到最终值 $\lambda_{998}(A) = -2.700\,001$.

373

同时，用蓝色表示的第四个最小的特征值 $\lambda_{997}(T_k)$ 误收敛于靠近 $\lambda_{996}(A) \approx -2.64$ 的值；在中间右边图像中的蓝点线指出靠近第 $k=61$ 步时 $\lambda_{997}(T_k)$ 和 $\lambda_{996}(A)$ 能达到9位小数位相同. 在第 $k=65$ 步重新开始迅速收敛到最终值 $\lambda_{997}(A) = -2.7$. 这也能在右下方图像中看出，其中 e_{997} 和 e_{998} 的特征向量分量在 $50<k<65$ 步之间再次增大，之后它们开始迅速收敛并且它们重新减小.

事实上，若 $\lambda_{997}(A)$ 确实是一个双重特征值，我们断言 T_k 决不会有两个而只有一个靠近特征值的值(按精确的算术运算). (证明见问题7-3). 在图7-5(见彩插)中说明这点，其中已经稍稍地修改 A 使它确实有两个等于 -2.7 的特征值. 注意无论何时都只出现 $\lambda_{998}(A)=\lambda_{997}(A) = -2.7$ 的一个近似.

幸亏有许多应用，在这些应用中不是求每个特征值的所有重合的拷贝而是求一个拷贝就足够了．另外，利用“块兰乔斯”算法重新获得重特征值是可能的．（见7.6节中引用的算法）．

考察图7-3的右上方图像中的其他特征值，我们看出误收敛是十分普遍的，正如由同类颜色加号的频繁的短水平线段指示的，它朝右下降到下面的较小的特征值．例如，第七个最小的特征值由 $\boldsymbol{T}_k$ 的第五个（黑色的）第六个（红色的）和第七个（绿色的）最小的特征征值在不同的兰乔斯步上良好地逼近．

这些误收敛现象说明为什么由定理7.2的第二部分提供的可以计算的误差界对检查收敛性是本质的[198]．若误差界是小的，即使一个特征值“失踪”，计算的特征值确实是某个特征值的一个好的近似值．◇

374

存在另一个由 Kaniel 和 Saad 给出的误差界，它阐明为什么发生误收敛．这个误差界依赖于初始向量 $\boldsymbol{q}_1$ 和要求的特征向量之间的夹角、里茨值和所要求的特征值．换言之，它依赖于计算期间未知的量，因而它是不实用的．但是它证明当 $\boldsymbol{q}_1$ 几乎正交于要求的特征向量时，或者当要求的特征值接近重特征值时，可以预期缓慢的收敛性．细节见[197，12.4节]．

7.4　浮点算术运算的兰乔斯算法

上节中的例子描述了本质上无舍入的“理想的”兰乔斯算法的性态．为了把它与原来的代价不昂贵的称为*非再正交化的兰乔斯算法*（HOMEPAGE/Matlab/LanczosNoReorthog. m）执行过程对比，我们称对应的精细而代价昂贵的算法6.10的执行过程为*完全再正交化的兰乔斯算法*．下面同时指出两个算法．

算法7.2　求 $A=A^{\mathrm{T}}$ 的特征值和特征向量的完全再正交化的或非再正交化的兰乔斯算法：

$\boldsymbol{q}_1=\boldsymbol{b}/\|\boldsymbol{b}\|_2, \beta_0=0,\ \boldsymbol{q}_0=0$

for $j=1$ to k

　　$z=\boldsymbol{A}\boldsymbol{q}_j$

　　$\alpha_j=\boldsymbol{q}_j^{\mathrm{T}}z$

　　$\begin{cases} z=z-\sum_{i=1}^{j-1}(z^{\mathrm{T}}\boldsymbol{q}_i)\boldsymbol{q}_i,\quad z=z-\sum_{i=1}^{j-1}(z^{\mathrm{T}}\boldsymbol{q}_i)\boldsymbol{q}_i & \text{完全再正交化} \\ z=z-\alpha_j\boldsymbol{q}_j-\beta_{j-1}\boldsymbol{q}_{j-1} & \text{非再正交化} \end{cases}$

　　$\beta_j=\|z\|_2$

　　if $\beta_j=0$, quit

　　$\boldsymbol{q}_{j+1}=z/\beta_j$

　　计算 $\boldsymbol{T}_k$ 的特征值，特征向量和误差界．

end for

为了使 z 几乎必定正交于 $\boldsymbol{q}_1$ 到 $\boldsymbol{q}_{j-1}$，完全再正交化对应于两次应用格拉姆-施密特正交化过程“$z=z-\sum_{i=1}^{j-1}(z^{\mathrm{T}}\boldsymbol{q}_i)\boldsymbol{q}_i$”.（为讨论何时“两次是足够的”见算法3.1以及[197，6-9节]，和[171，第7章].）在精确算术运算下，6.6.1节中指出 z 正交于 $\boldsymbol{q}_1$ 到 $\boldsymbol{q}_{j-1}$ 不要再正交化. 遗憾地，我们将看到舍入破坏迄今为止所有我们的分析所依赖的正交性性质.

正交性的丢失不会引起算法完全无法预言地运转. 事实上，将看到付出的代价是得到收敛的里茨值的*重复的拷贝*(multiple copies). 换言之，对大的 k 代替 T_k 有一个几乎等于 $\lambda_i(A)$ 的特征值，而几乎等于 $\lambda_i(A)$ 的特征值可能有许多. 若人们不涉及有关计算特征值的重数且不在乎内部的特征值产生延滞收敛，则这不是一个灾难. 详尽地描述以这种方式操作兰乔斯执行过程以及软件本身 NETLIB/lanczos 见[57].（这个软件具有估计特征值重数的启发式论据.） [375]

但是如果准确的重数是重要的话，则我们需要保持兰乔斯向量(几乎)正交. 因而可使用完全再正交化的兰乔斯算法，正如上节中所做的那样. 可以容易证明这个方法 k 步花费 $O(k^2n)$ flops 而不是 $O(kn)$ flops，以及 $O(kn)$ 空间而不是 $O(n)$ 空间，这代价可能太高以致无法承受.

幸运地，存在介于非再正交化和完全再正交化之间的中间地带，它几乎得到两个领域的最好的结果. 它通过发展在已经收敛的里茨向量的方向中的大的分量，q_k 以非常对称的方式丢失它们的正交性.（这是为什么导致收敛的里茨值的重复的拷贝.）通过下例以及下面 Paige 定理的说明来解释这个正交性的系统的降低. 通过监视计算的误差界将看到可以保守地预估哪个 q_k 将具有哪个里茨向量的大的分量. 于是可以仅仅以前面几个里茨向量为背景*有选择地正交化*(selectively orthogonalize) q_k，而不是像完全再正交化那样在每一步以所有的较早的 $\boldsymbol{q}_i$ 为背景. 这样以非常少的额外工作保持兰乔斯向量(几乎)正交. 下一节详细讨论选择正交化.

例 7.2 图 7-7(见彩插)表示对例 7.1 中的矩阵作兰乔斯算法 149 步的收敛性态. 右边的图像是完全再正交化而左边的图像是非再正交化. 除了省略整体误差外，这些图像类似于图 7-3 中的那些图像，因为整体误差把中间的图像弄得杂乱.

与兰乔斯第 k 步相对图 7-6 画出最小的奇异值 $\sigma_{\min}(\boldsymbol{Q}_k)$. 按精确算术运算，$\boldsymbol{Q}_k$ 正交因而 $\sigma_{\min}(\boldsymbol{Q}_k)=1$. 有舍入时，$\boldsymbol{Q}_k$ 大约在 $k=70$ 步处开始失去正交性，而 $\sigma_{\min}(\boldsymbol{Q}_k)$ 经过 $k=80$ 步下降至 .01，在图 7-7 中上面的两个图像在 $k=80$ 处开始看得出发散.

特别地，在图 7-7 的左上方图像中从第 $k=80$ 步开始，第二个最小的(红色的)特征值 $\lambda_2(\boldsymbol{T}_k)$ 已收敛到 $\lambda_2(\boldsymbol{A})\approx 2.7$ 几乎达到 16 位，只要再做几步就跳到 $\lambda_1(\boldsymbol{A})\approx 2.81$ 与 $\lambda_1(\boldsymbol{T}_k)$(黑色的)一起得到 $\lambda_1(\boldsymbol{A})$ 的第二个拷贝.（这可能是难以观察的. 因为红色的加号覆盖，因而遮掩了黑色的加号.）这个转变可在中左方图像中由红色虚线的误差界中的跳跃中看出. 此外，由左下方图像中 e_1 的递增的分量“预示”这个转变，其中黑色的曲线在第 $k=50$ 步开始再次上升而不是继续递减至机器精度，正如右下方图像中它用完全再正交化操作那样. 这两种图像都显示算法正在偏离它的精确的路 [376]

径(并且要求某一选择正交化). 在 $\lambda_1(A)$ 的第二个拷贝收敛之后, 兰乔斯向量中的 e_1 的分量开始再次下降, 其在第 $k=80$ 步之后, 才一点点开始下降.

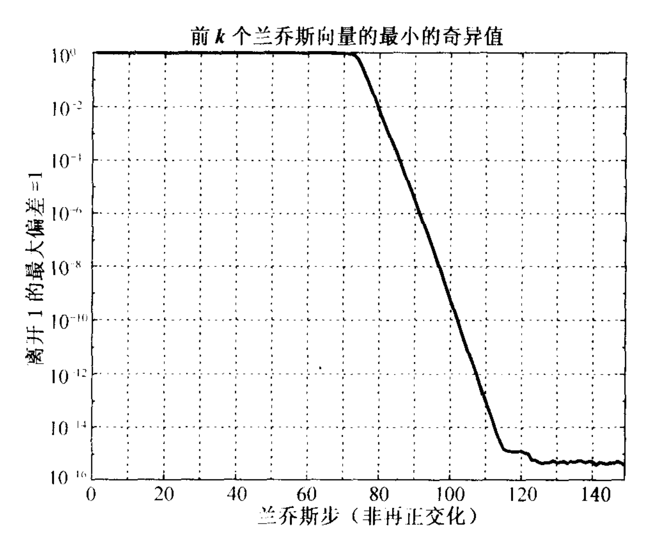


图7-6 非再正交化兰乔斯算法应用于 A. 对 $k=1$ 到 149 指出兰乔斯向量矩阵 Q_k 的最小奇异值 $\sigma_{\min}(Q_k)$. 在不出现舍入时, Q_k 正交, 而所有的奇异值应该是 1. 有舍入时, Q_h 变成秩亏阵.

类似地, 当左上方图像中的蓝色曲线($\lambda_4(T_k)$)从大约 $\lambda_3(A)\approx 2.6$ 移动到 $\lambda_2(A)\approx 2.7$时, $\lambda_2(A)$的第二个拷贝大约在第 $k=95$ 步开始出现. 此时我们有两个 $\lambda_1(A)\approx 2.81$ 的拷贝和两个 $\lambda_2(A)$的拷贝. 这在图像上面观察有一点困难, 因为一种颜色的加号遮掩其他颜色的加号(红色覆盖黑色, 而蓝色覆盖绿色). 由中左方图像中 $\lambda_4(T_k)$的蓝色虚线的误差界显示这个转变靠近 $k=95$ 时急剧地上升, 并由左下方图像中的上升的红色曲线预示这个转变. 它显示兰乔斯向量中的 e_2 的分量正在上升. 这个分量靠近 $k=95$ 时达到最高点并开始重新下降.

最后, 在第 $k=145$ 左右 $\lambda_1(A)$的第三个拷贝出现, 由两个左下方图像中的改变再次指示和预示. 若要继续兰乔斯过程, 则我们将周期地得到许多其他收敛的里茨值额外的拷贝. ◇

下面的定理对上例中看到的性态提供一个说明, 并且对选择正交化的兰乔斯向量含蓄地说到一个实用的准则. 为了不被所考虑的一切可能的舍入误差控制, 将利用他人的经验来识别那些少数几个重要的舍入误差并简单地略去其余的舍入误差 [197, 13-14 节]. 这使我们可把非再正交化的兰乔斯算法归结为一行.

[377]

$$\beta_j q_{j+1} + f_j = Aq_j - \alpha_j q_j - \beta_{j-1} q_{j-1}. \tag{7.3}$$

在此式中除了 f_j 外, 变量表示实际上存储在机器中的值. 而 f_j 表示由计算等式右边的值然后计算 β_j 和 q_{j+1}引起的舍入误差. 范数 $\|f_j\|_2$ 以 $O(\varepsilon\|A\|)$为界, 其中 ε 是机器

精度，这就是我们需要知道的有关 f_j 的全部说明. 另外，将精确地记 $\boldsymbol{T}_k = \boldsymbol{V\Lambda V}^{\mathrm{T}}$，因为我们知道在这个特征分解中出现的舍入误差是不重要的. 因而，$\boldsymbol{Q}_k$ 不一定是正交阵，但 $\boldsymbol{V}$ 是正交阵.

定理 7.3 Paige. 利用上一段中的记号和假设. 又设 $\boldsymbol{Q}_k = [\boldsymbol{q}_1, \cdots, \boldsymbol{q}_k]$，$\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k]$ 和 $\boldsymbol{\Lambda} = \mathrm{diag}(\theta_1, \cdots, \theta_k)$. 继续称 $\boldsymbol{Q}_k\boldsymbol{V}$ 的列 $y_{k,i} = \boldsymbol{Q}_k\boldsymbol{v}_i$ 为里茨向量和 θ_i 为里茨值. 则

$$\boldsymbol{y}_{k,i}^{\mathrm{T}} q_{k+1} = \frac{O(\varepsilon\|\boldsymbol{A}\|)}{\beta_k\,|\boldsymbol{v}_i(k)|}.$$

换言之，在里茨向量 $\boldsymbol{y}_{k,i} = \boldsymbol{Q}_k\boldsymbol{v}_i$ 方向中计算的兰乔斯向量 $\boldsymbol{q}_{k+1}$ 的分量 $\boldsymbol{y}_{k,i}^{\mathrm{T}}\boldsymbol{q}_{k+1}$ 与对应于里茨值 θ_i 的误差界 $\beta_k|v_i(\boldsymbol{k})|$ 的倒数成比例(见定理 7.2 的第 2 部分). 于是，当里茨值 θ_i 收敛并且它的误差界 $\beta_k|\boldsymbol{v}_i(k)|$ 趋于零时，兰乔斯向量 $\boldsymbol{q}_{k+1}$ 在里茨向量 $\boldsymbol{y}_{k,i}$ 方向中得到一个大的分量. 因而，正如例 7.2 中所见里茨向量变成线性相关. 确实，图 7-8 同时画出 1000×1000 对角阵例子最大的里茨值($i=1$，上方的图像)和第二个最大的里茨值($i=2$，下方的图像)的误差界 $|\beta_k\boldsymbol{v}_i(k)|/|\lambda_i(\boldsymbol{A})| \approx |\beta_k\boldsymbol{v}_i(k)|/\|\boldsymbol{A}\|$ 和里茨向量分量 $\boldsymbol{y}_{k,i}^{\mathrm{T}}\boldsymbol{q}_{k+1}$. 按照 Paige 定理，这两个量之积应为 $O(\varepsilon)$. 因为由这些半对数图像中线 $\sqrt{\varepsilon}$ 附近的曲线的对称性可以看出它确实如此.

Paige 定理的证明 对 $j=1$ 到 $j=k$，从(7.3)式开始，并把这 k 个式子记为单个式子

$$\boldsymbol{AQ}_k = \boldsymbol{Q}_k\boldsymbol{T}_k + [0, \cdots, 0, \beta_k\boldsymbol{q}_{k+1}] + \boldsymbol{F}_k = \boldsymbol{Q}_k\boldsymbol{T}_k + \beta_k\boldsymbol{q}_{k+1}\boldsymbol{e}_k^{\mathrm{T}} + \boldsymbol{F}_k,$$

378～379

其中 $\boldsymbol{e}_k^{\mathrm{T}}$ 是 k 维行向量 $[0, \cdots, 0, 1]$ 而 $\boldsymbol{F}_k = [f_1, \cdots, f_k]$ 是舍入误差矩阵. 去掉下标 k 简化记号得到 $\boldsymbol{AQ} = \boldsymbol{QT} + \beta\boldsymbol{qe}^{\mathrm{T}} + \boldsymbol{F}$. 用 $\boldsymbol{Q}^{\mathrm{T}}$ 左乘得到 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{AQ} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{QT} + \beta\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{qe}^{\mathrm{T}} + \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{F}$. 因为 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{AQ}$ 对称，得到 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{QT} + \beta\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{qe}^{\mathrm{T}} + \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{F}$ 等于它的转置或重新排列这个等式得到

$$0 = (\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{QT} - \boldsymbol{TQ}^{\mathrm{T}}\boldsymbol{Q}) + \beta(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{qe}^{T} - \boldsymbol{eq}^{\mathrm{T}}\boldsymbol{Q}) + (\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{F} - \boldsymbol{F}^{\mathrm{T}}\boldsymbol{Q}) \tag{7.4}$$

若 θ 和 v 分别是里茨值和里茨向量，则 $\boldsymbol{Tv} = \theta\boldsymbol{v}$，然后，注意

$$\boldsymbol{v}^{\mathrm{T}}\beta(\boldsymbol{eq}^{\mathrm{T}}\boldsymbol{Q})\boldsymbol{v} = [\beta\boldsymbol{v}(k)] \cdot [\boldsymbol{q}^{\mathrm{T}}(\boldsymbol{Qv})] \tag{7.5}$$

是误差界 $\beta v(k)$ 和里茨向量分量 $\boldsymbol{q}^{\mathrm{T}}(\boldsymbol{Qv}) = \boldsymbol{q}^{\mathrm{T}}\boldsymbol{y}$ 之积，Paige 定理宣称这个值应为 $O(\varepsilon\|\boldsymbol{A}\|)$. 现在我们的目标是巧妙地处理(7.4)式得到 $\boldsymbol{eq}^{\mathrm{T}}\boldsymbol{Q}$ 单独的一个表达式，然后利用(7.5)式.

为了这个目标，现在借助有关舍入的更简化的假设：因为 $\boldsymbol{Q}$ 的每列是通过用向量 z 的范数除 z 得到的，所以 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}$ 的对角元等于 1 达到全机器精度；我们将假定它精确地等于 1. 而且通过构造兰乔斯算法计算的向量 $z' = z - \alpha_j\boldsymbol{q}_j = z - (\boldsymbol{q}_j^{\mathrm{T}}z)\boldsymbol{q}_j$ 正交于 $\boldsymbol{q}_j$，使得 $\boldsymbol{q}_{j+1}$ 和 $\boldsymbol{q}_j$ 的正交几乎达到全机器精度的程度. 故 $\boldsymbol{q}_{j+1}^{\mathrm{T}}\boldsymbol{q}_j = (\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q})_{j+1,j} = O(\varepsilon)$；我们将简单地假设 $(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q})_{j+1,j} = 0$. 现在记 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \boldsymbol{I} + \boldsymbol{C} + \boldsymbol{C}^{\mathrm{T}}$，其中 $\boldsymbol{C}$ 是下三角阵. 因为有关舍入的假设，事实上 $\boldsymbol{C}$ 仅在第二条次对角线上以及下面是非零的. 这意味着.

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{QT} - \boldsymbol{TQ}^{\mathrm{T}}\boldsymbol{Q} = (\boldsymbol{CT} - \boldsymbol{TC}) + (\boldsymbol{C}^{\mathrm{T}}\boldsymbol{T} - \boldsymbol{TC}^{\mathrm{T}}),$$

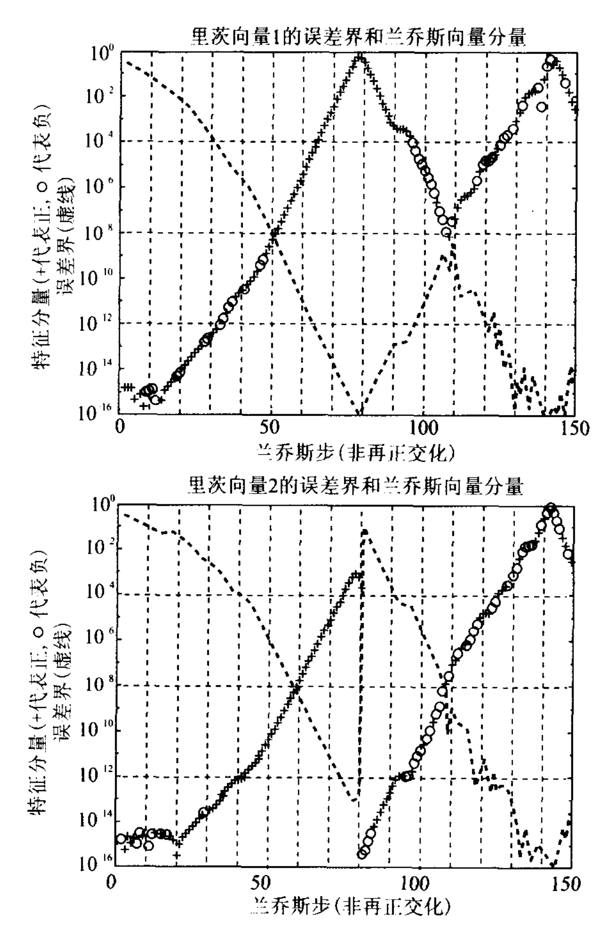


图 7-8　用非再正交化兰乔斯算法于 $\boldsymbol{A}$. 对最大的特征值（上面的）和第二最大的特征值（下面的）指出前 149 步. 如前面一样虚线是误差界. 用 + 和○标记的线表示在最大的里茨值（$i=1$，在上面）或第二最大的里茨值（$i=2$，在下面）的里茨向量方向中兰乔斯向量的第 $k+1$ 个分量 $\boldsymbol{y}_{k,i}^{\mathrm{T}}\boldsymbol{q}_{k+1}$

我们可利用 $\boldsymbol{C}$ 和 $\boldsymbol{T}$ 的零结构，容易证明 $\boldsymbol{CT}-\boldsymbol{TC}$ 是严格下三角阵而 $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{T}-\boldsymbol{TC}^{\mathrm{T}}$ 是严格上三角阵. 此外，因为 $\boldsymbol{e}$ 仅在它的最后一个元素非零，所以 $\boldsymbol{eq}^{\mathrm{T}}\boldsymbol{Q}$ 仅在它的最后行非零. 而且，刚才描述的 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}$ 的结构推出 $\boldsymbol{eq}^{\mathrm{T}}\boldsymbol{Q}$ 最后行的最后元素为零. 故特别地，$\boldsymbol{eq}^{\mathrm{T}}\boldsymbol{Q}$ 也是严格下三角阵且 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{qe}^{\mathrm{T}}$ 是严格上三角阵. 应用 $\boldsymbol{eq}^{\mathrm{T}}\boldsymbol{Q}$ 和 $\boldsymbol{CT}-\boldsymbol{TC}$ 两者都是严格下三角阵的事实于(7.4)式得到

$$0=(\boldsymbol{CT}-\boldsymbol{TC})-\beta\boldsymbol{eq}^{\mathrm{T}}\boldsymbol{Q}+\boldsymbol{L}, \tag{7.6}$$

其中 $\boldsymbol{L}$ 是 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{F}-\boldsymbol{F}^{\mathrm{T}}\boldsymbol{Q}$ 的严格下三角部分. 用 $\boldsymbol{v}^{\mathrm{T}}$ 左乘和用 $\boldsymbol{v}$ 右乘(7.6)式，利用(7.5)式和 $\boldsymbol{v}^{\mathrm{T}}(\boldsymbol{CT}-\boldsymbol{TC})\boldsymbol{v}=\boldsymbol{v}^{\mathrm{T}}\boldsymbol{Cv}\theta-\theta\boldsymbol{v}^{\mathrm{T}}\boldsymbol{Cv}=0$，得到

$$v^{\mathrm{T}}\beta(eq^{\mathrm{T}}Q)v=[\beta v(k)]\cdot[q^{\mathrm{T}}(Qv)]=v^{\mathrm{T}}Lv.$$

因为 $|v^{\mathrm{T}}Lv|\leqslant\|L\|=O(\|Q^{\mathrm{T}}F-F^{\mathrm{T}}Q\|)=O(\|F\|)=O(\varepsilon\|A\|)$，所以我们得到

380 ~ 381

$$[\beta v(k)]\cdot[q^{\mathrm{T}}(Qv)]=O(\varepsilon\|A\|),$$

这等价于 Paige 定理. □

7.5 选择正交化的兰乔斯算法

本节讨论兰乔斯算法的一种变形，它(几乎)具有完全再正交化兰乔斯算法的高精度但(几乎)具有非再正交化兰乔斯算法的低成本. 这个算法称为选择正交化(selective orthogonalization)兰乔斯算法. 正如上节中讨论的那样，我们的目标是通过每一步把相对于 q_k 的其他尽可能少数的向量正交化(低成本)保持计算的兰乔斯向量 q_k 尽可能几乎正交(高精度). Paige 定理(上节中定理 7.3)告诉我们 q_k 失去正交性是因为里茨值 θ_i 收敛的里茨向量 $y_{i,k}=Q_kv_i$ 方向中它们得到大的分量，θ_i 的收敛以变小的误差界 $\beta_k|v_i(k)|$来度量. 这个现象在例 7.2 中说明.

因而，选择正交化的最简单的形式仅仅在每一步测试误差界 $\beta_k|v_i(k)|$，当它变得足够小时，兰乔斯算法内循环中的向量 z 相对于 $y_{i,k}$ 正交：$z=z-(y_{i,k}^{\mathrm{T}}z)y_{i,k}$. 当它小于 $\sqrt{\varepsilon}\|A\|$ 时认为 $\beta_k|v_i(k)|$是小的，因为 Paige 定理告诉我们向量分量 $|y_{i,k}^{\mathrm{T}}q_{k+1}|=|y_{i,k}^{\mathrm{T}}z/\|z\|_2|$ 可能超过 $\sqrt{\varepsilon}$. (实际上可用 $\|T_k\|$ 代替 $\|A\|$，因为 $\|T_k\|$ 已知而 $\|A\|$ 可能不知道.)这就导致下列算法.

算法 7.3 求 $A=A^{\mathrm{T}}$ 的特征值和特征向量的选择正交化兰乔斯算法：

> $q_1=b/\|b\|_2$， $\beta_0=0$，$q_0=0$
> for $j=1$ to k
> $z=Aq_j$
> $\alpha_j=q_j^{\mathrm{T}}z$
> $z=z-\alpha_jq_j-\beta_{j-1}q_{j-1}$
>   /* 对收敛的里茨向量选择正交化 */
>   for all $i\leqslant k$ such that $\beta_k|v_i(k)|\leqslant\sqrt{\varepsilon}\|T_k\|$
> $z=z-(y_{i,k}^{\mathrm{T}}z)y_{i,k}$
>   end for
> $\beta_j=\|z\|_2$
>   if $\beta_j=0$, quit
> $q_{j+1}=z/\beta_j$
>   计算 T_k 的特征值，特征向量和误差界
> end for

下例表明对早先的 1000×1000 阶对角阵使用这个算法(HOMEPAGE/Matlab/
382 Lanczos SelectOrthog. m)时发生什么情况.

例 7.3　选择正交化的兰乔斯算法的性态与图 7-7 右面的三个图像中指出的完全正交化的兰乔斯算法的性态表面上是不能区别的. 换言之，选择正交化提供与完全正交化差不多的精度.

所有 $\boldsymbol{Q}_k$ 的最小奇异值大于 $1-10^{-8}$，这意味着正如要求的那样选择正交化保持兰乔斯向量正交达到大约一半精度.

图 7-9(见彩插)表示对应于再正交化选择的里茨向量的里茨值. 因为对应于收敛的里茨值选择的里茨向量和最大的与最小的里茨值首先收敛，所以有两个图像：大的收敛的里茨值在上面，而小的收敛的里茨值在下面. 上面的图像匹配图 7-7 中右上方的图像中指出的已收敛达到至少一半精度的里茨值. 对全部 $149 \times 150/2 = 11\ 175$ 种可能的正交总共选择 1485 个里茨向量. 因而选择正交化仅仅做了差不多是完全再正交化保持兰乔斯向量(几乎)正交的再正交化工作的 $1485/11\ 175 \approx 13\%$.

图 7-10(见彩插)表示选择再正交化兰乔斯算法如何只对两个最大的里茨值的里茨向量保持兰乔斯向量正交. 上方的图像是图 7-8 中两个图像的叠加，它表示非再正交化的兰乔斯算法的误差界和里茨向量分量. 下方的图像是选择正交化兰乔斯算法对应的图像. 注意在第 $k=50$ 步处最大的特征值的误差界(黑虚线)达到阈值 $\sqrt{\varepsilon}$. 选择里茨向量正交化(由图 7-9 上方图中最高的黑色加号表示)，在这个里茨向量方向中的分量从图 7-10 的下方图像中消失. 之后若干步，在第 $k=58$ 步处第二个最大的里茨值的误差界达到 $\sqrt{\varepsilon}$，并且它也被选择正交化. 在下方图像中的误差界继续递减至机器精度 ε 并停留在那里，但是上方图像中的误差界最终再次增长. ◇

7.6　选择正交化之外的方法

选择正交化并非讨论课题的结束，因为即使代价稍昂贵，也可以构造对称的兰乔斯算法. 结果是一旦一个兰乔斯向量已相对于一个特殊的里茨向量 $\boldsymbol{y}$ 正交，在兰乔斯向量再次需要对 $\boldsymbol{y}$ 正交之前它作了许多步. 故算法 7.3 中许多正交化工作可以省略. 事实上，有一个决定何时再正交化的一个简单而便宜的递归[224,192]. 另一种
383 提高是使用误差界有效地区分收敛的和“误收敛”特征值[198]. 达到最新发展水平的兰乔斯算法执行过程在[125]中描述. 一个不同的软件执行过程在 ARPACK (NETLIB/scalapack/readme. arpack[171,233])中可以得到.

若对位移和求逆矩阵 $(\boldsymbol{A}-\sigma\boldsymbol{I})^{-1}$ 应用兰乔斯算法，则我们预期最接近 σ 的特征值首先收敛. 有一些不同的“预条件”矩阵 $\boldsymbol{A}$ 更快地收敛到某些特征值的方法. 例如，Davidson 方法[60]用于量子化学问题，其中 $\boldsymbol{A}$ 是强对角占优的. 把 Davidson 方法与雅可比法组合起来也是可能的[229].

7.7 非对称特征值问题的迭代算法

当A非对称时，上面描述的兰乔斯算法不能再应用. 有两个可供选择的办法. 384

第一个可供选择的办法是使用Arnoldi算法(算法6.9). 记得阿诺尔迪算法计算克雷洛夫子空间$\mathcal{K}_k(A,q_1)$的一个正交基Q_k使得$Q_k^TAQ_k=H_k$是上海森伯格而不是对称三对角阵. 类比瑞利-里茨过程，再次用H_k的特征值近似A的特征值. 因为A非对称，所以它的特征值可能是复的和/或坏条件的，使得兰乔斯算法享有的并在7.3节中描述的许多吸引人的误差界和单调收敛性质不再成立. 但是，存在一些有效的算法和执行过程. 有益的参考文献包含在[154,171,212,216,217,233]和著作[213]中. 最新的软件在[171,233]中描述并且可以在NETLIB/scalapack/readme. arpack中找到. Matlab指令eigs(关于"稀疏特征值")使用这个软件.

第二个可供选择的办法是使用*非对称兰乔斯算法*. 这个算法试图通过非正交相似把A化为非对称三对角形式. 期待求(稀疏的!)非对称三对角阵的特征值比用阿诺尔迪算法求海森伯格阵的特征值更容易. 遗憾的是，相似变换可能十分病态，这意味着三对角阵的特征值和原矩阵的特征值可能差别很大. 事实上，由于大家知道的"故障"(breakdown)现象[42,134,135,199]不一定可能找一个适当的相似. 尝试用一个称为"先行"(look-ahead)的过程修理故障，这个过程在[16,18,55,56,64,108,202,265,266]中提出并作了分析.

最后，对稀疏非对称特征问题应用子空间迭代(算法4.3)[19]，Davidson算法[216]，或者雅可比-Davidson算法[230]是可能的.

7.8 第7章的参考书目和其他话题

除7.6节和7.7节的参考书目外，关于稀疏特征值问题算法有许多好的综述：见[17.51,125,163,197,213,262]. 在[76]中也讨论并行执行过程.

在6.2节中我们讨论存在在线(On-line)帮助从各种迭代法中选择可利用的方法求解$Ax=b$. 关于特征问题的一个类似的科研项目正在进行中，并且将把它收入到本书未来的版本中.

7.9 第7章问题

问题7.1(容易) 证实对A用初始向量q与对Q^TAQ用初始向量Q^Tq一样运行阿诺尔迪算法(算法6.9)或兰乔斯算法(算法6.10)得到相同的三对角阵T_k(或海森伯格阵H_k). 385~386

问题7.2(中等) 设λ_i是A的单特征值. 证实若q_1正交于A对应的特征向量，

则在被计算的最大的 T_k 不会把 λ_i 作为特征值的意义下，通过精确算术运算的兰乔斯算法计算的三对角阵 T_k 的特征值不会收敛于 λ_i. 利用 3×3 例子表明某个其他的 T_k 的特征值可能“意外地”等于 λ_i.

问题7.3(中等) 证实用兰乔斯算法计算的对称三对角阵 T_k 不会精确地有重特征值. 证明若 A 有重特征值，则对 A 应用兰乔斯算法一定无法进行到最后一步.

387

索　　引

索引中页码为英文原书页码，与书中页边标注的页码一致.

D

K

L

M

N

O

P

S

T